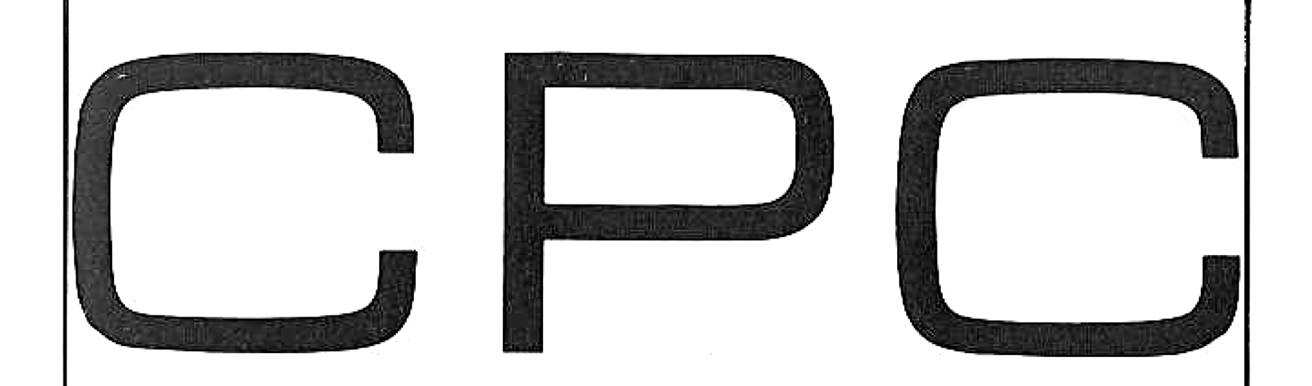
Informatique AMSTRAD

CPC

COMPILATION DES NUMEROS 1··2··3··4







COMPILATION

DES

NUMEROS

1··2··3··4



La Haie de Pan – 35170 BRUZ

1

LE MANOIR DE ROCHEBRUNE

Pascal et Laurent KERLOCH

En quête d'une aventure, arrivez dans le pays de Dragonnaze qui domine une solide forteresse. Le Baron de ce territoire a jeté depuis bien longtemps un maléfice sur la région. De plus, il retient sa fille, la belle Hélène, prisonnière dans son donjon, la Tour Noire.

De nombreux princes du royaume sont venus lui demander sa main, mais aucun n'a pu ressortir vivant du château. En apprenant cela, vous décidez, vous aussi, de tenter votre chance.
Pour obtenir l'accès à la Tour Noire et délivrer la Princesse, il vous faudra réunir un diamant, un lingot, un coffre, un sceptre et une couronne, ainsi que tuer le Dragon (mais attention ! ses coups sont mortels !).
BONNE CHANCE !

```
10 '        * * * * * * * * * * *
20 '        *                   *
30 '        *     LE MANOIR     *
40 '        *                   *
50 '        *         DE        *
60 '        *                   *
70 '        *     ROCHEBRUNE    *
80 '        *                   *
90 '        * * * * * * * * * * *
100 '
110 '            par Laurent
120 '
130 '            et Pascal   Kerloch
140 '
150 '
160 '            Copyright 1985
170 '
```

```
180 ' ****************
190 ' * PRESENTATION *
200 ' ****************
210 MODE 1:BORDER 1
220 INK 0,0:INK 1,18:INK 2,6:INK 3,24
230 PEN 2: LOCATE 16,5:PRINT"Le Manoir"
240 LOCATE 20,8:PRINT"de"
250 LOCATE 16,11:PRINT"Rochebrune"
260 PEN 3
270 LOCATE 16,6:PRINT STRING$(10,CHR$(208))
280 LOCATE 20,9:PRINT STRING$(2,CHR$(208))
290 LOCATE 16,12:PRINT STRING$(10,CHR$(208))
300 PEN 1:LOCATE 20,15:PRINT"par Pascal"
310 PEN 1:LOCATE 20,17:PRINT"et Laurent Kerloch"
320 ENV 1,100,-0.5,20
330 ENV 2,50,-0.5,10
340 SOUND 1,239,60,12,1:SOUND 2,478,60,12,1
350 SOUND 1,319,30,12,2:SOUND 2,638,30,12,2
360 SOUND 1,319,30,12,2:SOUND 2,638,30,12,2
370 SOUND 1,239,60,12,1:SOUND 2,478,60,12,1
380 SOUND 1,190,60,12,1:SOUND 2,379,60,12,1
390 SOUND 1,169,60,12,1:SOUND 2,338,60,12,1
400 SOUND 1,190,30,12,2:SOUND 2,379,30,12,1
410 SOUND 1,190,30,12,2:SOUND 2,379,30,12,1
420 SOUND 1,169,60,12,1:SOUND 2,338,60,12,1
430 SOUND 1,190,60,12,1:SOUND 2,379,60,12,1
440 SOUND 1,169,120,12,1:SOUND 2,338,120,12,1
450 FOR ATT=1 TO 3000:NEXT ATT
460 '    ******************
470 '    * INITIALISATION *
480 '    ******************
490 MODE 1
500 ROUTI=1:PARAM=1:DIM ARM$(12):DIM TRES$(12)
510 DATA Diamant,Lingot,Coffre,Sceptre,Couronne,Dague,Arc,Hache,Armure,La
nce,Epee
520 FOR A=8 TO 12:READ A$:TRES$(A)=A$:NEXT A:FOR A=1 TO 12 STEP 2:READ A$
:ARM$(A)=A$:ARM$(A+1)=A$:NEXT A
530 CLS:DIM ARM(120):DIM TRES (12):WINDOW #1,28,40,1,2:WINDOW #2,28,40,4,
17:WINDOW #3,28,40,19,25:WINDOW #0,1,26,4,25:WINDOW #4,1,3,1,2:WINDOW #5,
5,26,1,2
540 INK 0,0:INK 1,15:INK 2,6:INK 3,18:FOR A=0 TO 5:PAPER #A,0:NEXT A:PEN
#0,1:PEN #1,1:PEN #2,3:PEN #3,2:PEN #4,3:PEN #5,2
```

```
550 BORDER 3
560 PLOT 50,400,2:DRAWR 0,-34,2:DRAWR -50,0,2:PLOTR 0,-12,2:DRAWR 418,0,2
:DRAWR 0,-354,2:PLOTR 12,0,2:DRAWR 0,114,2:DRAWR 210,0,2:PLOTR 0,12,2:DRA
WR -210,0,2:DRAWR 0,228,2:DRAWR 210,0,2:PLOTR 0,12,2:DRAWR -210,0,2:DRAWR
 0,34,2:PLOTR -12,0,2:DRAWR 0,-34,2
570 DRAWR -358,0,2:DRAWR 0,34,2
580 PRINT" Des rumeurs circulent quidisent que le terrible   Baron de Ro
chebrune garde sa fille, la Princesse   Helene, dans le Donjon de son Ma
noir, la Tour Noire "
590 PRINT" Vous avez decide de la  delivrer et vous vous    dirigez ver
s le sombre   Manoir de Rochebrune.":PRINT
600 IF INKEY$<>"" THEN 610 ELSE 600
610 CLS:PRINT" Le Sphinx qui en garde  l'entree vous enonce vos  droits
et obligations :":PRINT
620 PRINT" Vous devez rassembler 5  Tresors differents et tuerle Mechant
Baron qui des  votre arrivee se metamor- phose en un Terrifiant   Dragon
."
630 PRINT:PRINT"Mais vous etes bien decidea tenter votre chance et  vous
penetrez dans le Ma- noir. Le pont-levis se   referme derriere vous.":PR
INT
640 IF INKEY$="" THEN 640 ELSE CLS
650 ' ********************************
660 ' * TIRAGE DE LA FORCE DE DEPART *
670 ' ********************************
680 PRINT #1,"Votre Force"
690 FRC=INT(RND(1)*10)+2
700 IF FRC=2 THEN PRINT"Vous n'avez pas de chance du tout, l'ordinateur n
e  vous attribut que 2 pointsde valeur"
710 IF FRC>2 AND FRC<=5 THEN PRINT"Vous n'avez pas beaucoup":PRINT"de cha
nce, l'ordinateur":PRINT"vous donne "FRC;" points de":PRINT"valeur"
720 IF FRC>5 AND FRC<10 THEN PRINT"Vous avez de la chance,   l'ordinateur
 vous donne   ";FRC;" points de valeur"
730 IF FRC>=10 AND FRC<=12 THEN PRINT"L'ordinateur vous apprecie";"vraime
nt beaucoup, il vous";"donne"FRC"points de valeur"
740 PRINT
750 ' *************
760 ' * RENCONTRE *
770 ' *************
780 FOR ATT=0 TO 3000:NEXT ATT:CLS
790 CLS #5:CLS #4
800 LOCATE #1,1,2:PRINT #1,"    "FRC
810 GOSUB 880
```

```
820 IF FRC<0 THEN 3900
830 RC=INT(RND(1)*100)+1
840 IF RC<=40 THEN 1050
850 IF RC<=65 THEN 2100
860 IF RC<=90 THEN 3130
870 IF RC<=100 THEN 2380
880 '            TABLEAU DES ARMES
890 LOCATE #2,1,1
900 PRINT#2,"Armes :":PRINT#2,""
910 FOR D=1 TO 120
920 IF ARM(D)=0 THEN 950
930 IF ARM(D)=12 THEN PRINT #2,"Epee de Feu":GOTO 950
940 PRINT #2,ARM$(ARM(D));TAB(9)ARM(D)
950 NEXT D
960 '            TABLEAU DES TRESORS
970 LOCATE #3,1,1
980 PRINT#3,"Tresors :":PRINT#3,""
990 FOR D=1 TO 6
1000 IF TRES(D)=0 THEN 1020
1010 PRINT #3,TRES$(TRES(D));TAB(9)TRES(D)
1020 NEXT D
1030 IF ROUTI=6 AND DRAGON=1 THEN 4090
1040 RETURN
1050 ' *****************************
1060 ' * RENCONTRE AVEC UN MONSTRE *
1070 ' *****************************
1080 '    TYPE DE MONSTRE
1090 TDM=INT(RND(1)*100)+1
1100 IF TDM<=2 THEN M$="e Licorne":B=5:GOTO 1300
1110 IF TDM<=5 THEN M$=" Centaure":B=8:GOTO 1300
1120 IF TDM<=8 THEN M$=" Spectre":B=4:GOTO 1300
1130 IF TDM<=10 THEN M$=" Gobelin":B=3:GOTO 1300
1140 IF TDM<=13 THEN M$=" Squelette":B=4:GOTO 1300
1150 IF TDM<=16 THEN M$="e Sorciere":B=5:GOTO 1300
1160 IF TDM<=19 THEN M$=" Sphinx":B=6:GOTO 1300
1170 IF TDM<=21 THEN M$="e Chimere":B=8:GOTO 1300
1180 IF TDM<=24 THEN M$=" Titan":B=10:GOTO 1300
1190 IF TDM<=27 THEN M$=" Demon":B=6:GOTO 1300
1200 IF TDM<=30 THEN M$=" Fantome":B=6:GOTO 1300
1210 IF TDM<=32 THEN M$="e Momie":B=5:GOTO 1300
1220 IF TDM<=35 THEN M$=" Diable":B=5:GOTO 1300
1230 IF TDM<=38 THEN M$=" Griffon":B=6:GOTO 1300
```

```
1240 IF TDM<=47 THEN M$=" Cobra":B=5:GOTO 1300
1250 IF TDM<=57 THEN M$=" Troll":B=9:GOTO 1300
1260 IF TDM<=66 THEN M$="e Goule":B=2:GOTO 1300
1270 IF TDM<=76 THEN M$=" Vampire":B=6:GOTO 1300
1280 IF TDM<=85 THEN M$=" Gnome":B=5:GOTO 1300
1290 IF TDM<=100 THEN M$=" Balrog":B=8:GOTO 1300
1300 '    FORCE DU MONSTRE
1310 FDM=B+(INT(RND(1)*3))
1320 PRINT"Vous etes attaque par":PRINT"un"M$" de force"FDM
1330 PRINT #5,"Monstre :"
1340 PRINT #5,"Un"M$" Force"FDM
1350 PRINT
1360 SOUND 1,319,16,5:SOUND 1,239,16,5:SOUND 1,253,16,5:SOUND 1,284,16,5
1370 SOUND 1,253,48,5:SOUND 1,239,16,5:SOUND 1,213,48,5:SOUND 1,190,16,5
1380 SOUND 1,213,16,5:SOUND 1,239,16,5:SOUND 1,253,16,5:SOUND 1,239,32,5
1390 SOUND 1,253,16,5:SOUND 1,284,48,5:SOUND 1,319,48,5
1400 REM    **********
1410 REM    * COMBAT *
1420 REM    **********
1430 INPUT"Voulez-vous combattre ce  monstre (O/N) ",DESIR$
1440 IF DESIR$="O" OR DESIR$="o" THEN 1460
1450 IF DESIR$="N" OR DESIR$="n" THEN 2050 ELSE 1430
1460 '    *******************
1470 '    * CHOIX DE L'ARME *
1480 '    *******************
1490 PRINT
1500 PRINT "Avec quelle arme voulez-  vous combattre ?":INPUT "  ( Si vou
s n'avez pas      d'arme tapez 0 )";VARM
1510 FOR V=0 TO 120
1520 IF ARM(V)=VARM OR VARM=0 THEN ABUS=0 : GOTO 1570
1530 NEXT V
1540 PRINT"Vous ne possedez pas cette arme ":ABUS=ABUS+1
1550 IF ABUS=10 THEN VARM=0:GOTO 1570
1560 GOTO 1460
1570 '    ********************
1580 '    * CALCUL DU COMBAT *
1590 '    ********************
1600 PRINT
1610 PRINT"Pour engager le combat    tapez sur une touche":PRINT
1620 CA=CA+1
1630 IF INKEY$<>"" THEN 1650
1640 GOTO 1620
```

```
1650 VCOMB=FDM-VARM
1660 IF VARM>FDM THEN VCOMB=0
1670 PRINT"Le combat est de force"VCOMB
1680 LOCATE #4,1,1:PRINT#4,"Des"
1690 FOR A=1 TO 20:SOUND 1,60,1,2
1700 DES=INT(RND(1)*10)+2
1710 DES$=(RIGHT$(STR$(DES),2))
1720 LOCATE #4,1,2:PRINT #4,DES$
1730 FOR B=1 TO 100:NEXT B
1740 NEXT A:SOUND 1,119,5,2
1750 IF DES<VCOMB THEN 1910
1760 IF DES>VCOMB THEN 1780
1770 IF DES=VCOMB THEN 1850
1780 '          *********
1790 '          * GAGNE *
1800 '          *********
1810 FRC=FRC+VCOMB
1820 PRINT"Vous avez gagne, les des  ont fait "DES
1830 PRINT
1840 GOTO 750
1850 '          *******************
1860 '          * LE MONSTRE FUIT *
1870 '          *******************
1880 PRINT"Le monstre fuit, les des  ont fait "DES
1890 PRINT
1900 GOTO 750
1910 '          *********
1920 '          * PERDU *
1930 '          *********
1940 FRC=FRC-VCOMB
1950 PRINT"Vous avez perdu, les des  ont fait "DES
1960 PRINT
1970 IF VARM=0 THEN 750
1980 FOR D=1 TO 120
1990 IF ARM(D)<>VARM THEN NEXT D
2000 ARM(D)=0
2010 NbArmes=NbArmes-1
2020 CLS #2
2030 GOSUB 880
2040 GOTO 750
```

```
2050 '    *********************
2060 '    * RETRAIT DE POINTS *
2070 '    *********************
2080 FRC=FRC-FDM
2090 GOTO 750
2100 ' ****************************
2110 ' * RENCONTRE AVEC UN TRESOR *
2120 ' ****************************
2130 VTRES=INT(RND(1)*5)+8
2140 IF RIGHT$(TRES$(VTRES),2)="ne" THEN T$="e " ELSE T$=" "
2150 PRINT"Vous avez trouve un"T$:PRINT TRES$(VTRES)" de valeur"VTRES
2160 PRINT
2170 SOUND 1,71,24,5:SOUND 1,60,48,5:SOUND 1,53,24,5:SOUND 1,47,98,5
2180 SOUND 1,53,24,5:SOUND 1,60,24,5:SOUND 1,53,24,5:SOUND 1,60,24,5
2190 SOUND 1,63,24,5:SOUND 1,71,120,5
2200 '          *********************
2210 '          * ACHAT DU TRESOR ? *
2220 '          *********************
2230 IF ROUTI=6 THEN PRINT"Mais vous avez deja les 5 tresors":PRINT:GOTO
750
2240 FOR D=1 TO 5
2250 IF TRES(D)=VTRES THEN PRINT"Mais vous possedez deja untresor de cett
e valeur":PRINT:GOTO 750
2260 NEXT D
2270 IF FRC<VTRES THEN PRINT"Mais ce tresor est trop   cher pour votre bo
urse":PRINT:GOTO 750
2280 INPUT "Voulez-vous l'acheter ?   (O/N) ";DESIR$
2290 IF DESIR$="O" OR DESIR$="o" THEN 2310
2300 IF DESIR$="N" OR DESIR$="n" THEN 750 ELSE 2280
2310 '          *******************************
2320 '          * CALCUL DE L'ACHAT DU TRESOR *
2330 '          *******************************
2340 TRES(ROUTI)=VTRES
2350 ROUTI=ROUTI+1
2360 FRC=FRC-VTRES
2370 GOTO 750
2380 ' *****************************
2390 ' * RENCONTRE AVEC LE SORCIER *
2400 ' *****************************
2410 PRINT"Vous rencontrez le sorcieret il vous propose son    aide en vo
us offrant un   sortilege."
2420 GOSUB 2490
```

```
2430 SOUND 1,239,64,5:SOUND 1,284,32,5:SOUND 1,284,48,5: SOUND 1,301,16,5

2440 SOUND 1,284,32,5:SOUND 1,253,64,5:SOUND 1,301,32,5:SOUND 1,379,64.5
2450 GOSUB 2490
2460 SOUND 1,239,48,5:SOUND 1,253,16,5:SOUND 1,284,32,5:SOUND 1,301,48,5
2470 SOUND 1,338.16,5:SOUND 1,301,32,5:SOUND 1,284,128.5
2480 GOTO 2530
2490 SOUND 1,284,32,5: SOUND 1,239,64,5:SOUND 1,213,32,5:SOUND 1.190,48,5
2500 SOUND 1,179,16,5:SOUND 1,190,32,5:SOUND 1,213,64,5 :SOUND 1,253,32,5
2510 SOUND 1,319,48,5:SOUND 1,284,16,5:SOUND 1,253,32,5
2520 RETURN
2530 '    ****************
2540 '    * CONSULTATION *
2550 '    ****************
2560 INPUT"Acceptez-vous ? (O/N) ",REP$
2570 IF REP$="O" OR REP$="o" THEN 2600
2580 IF REP$="N" OR REP$="n" THEN FOR ATT=0 TO 1500:NEXT ATT:CLS:GOTO 105
0
2590 GOTO 2560
2600 '    *************
2610 '    * SORTILEGE *
2620 '    *************
2630 SORT=INT(RND(1)*100)+1
2640 IF SORT<=15 THEN 2730
2650 IF SORT>15 AND SORT<=45 THEN 2790
2660 IF SORT>45 AND SORT<=70 THEN 2830
2670 IF SORT>70 AND SORT<=75 THEN 2940
2680 IF SORT>75 AND SORT<=100 THEN 2690
2690 ROUTI=ROUTI-1:IF ROUTI=0 THEN ROUTI=1:GOTO 2630
2700 TRES(ROUTI)=0
2710 PRINT:PRINT" Le Sorcier vous retire un Tresor.":CLS #3:GOSUB 880
2720 GOTO 2960
2730 ARM(PARAM)=12
2740 PARAM=PARAM+1
2750 NbArmes=NbArmes+1
2760 GOSUB 880
2770 IF NbArmes<13 THEN PRINT:PRINT"  Le Sorcier vous fait donde l'Epee d
e feu grace a  laquelle vous pourrez tuerle Dragon" ELSE 2960
2780 GOTO 2960
2790 FRC=FRC+12
2800 PRINT:PRINT" Le Sorcier vous offre unefiole qui contient un ele-xir.
 Grace a cette potion vous gagnez 12 points de  Force."
```

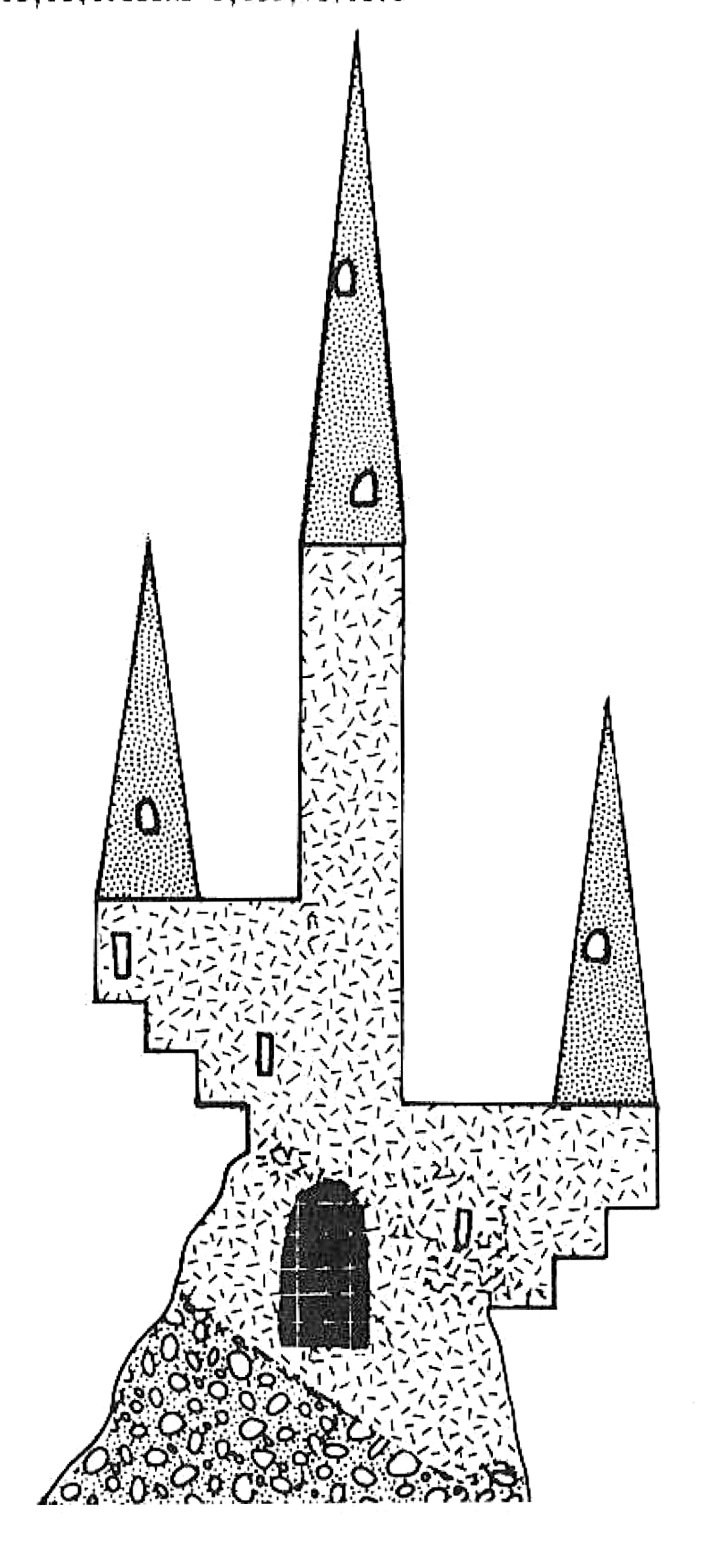
```
2810 LOCATE #1,1,2:PRINT #1,"    "FRC
2820 GOTO 2960
2830 PRINT:PRINT"Le Sorcier vous offre deuxarmes de votre choix":PRINT
2840 FOR E=1 TO 2
2850 IF NbArmes>=12 THEN PRINT" Mais vous possedez deja  12 armes.":GOTO
2960
2860 INPUT"  Une arme de quelle Puis-sance desirez-vous ?",VARM
2870 IF VARM<2 OR VARM>11 THEN 2860
2880 ARM(PARAM)=VARM
2890 PARAM=PARAM+1
2900 NbArmes=NbArmes+1
2910 GOSUB 880:PRINT
2920 NEXT E
2930 GOTO 2960
2940 IF DRAGON=1 THEN 2600 ELSE PRINT:PRINT"Le Sorcier tue le Dragon avot
re place, vous n'aurez donc pas a le faire"
2950 DRAGON=1
2960 '   *************************
2970 '   * COMBAT MONSTRE/DRAGON *
2980 '   *************************
2990 FOR ATT=0 TO 3000:NEXT ATT:CLS
3000 IF DRAGON=1 THEN 1050
3010 I=INT(RND(1)*2)+1
3020 IF I=1 THEN 1050
3030 ENV 1,6,4,5,9,-2,4,13,7,6,8,-1,5,3,7,8
3040 ENV 2,12,-2,6,13,5,6,8,-1,5,5,7,7,8,-4,9
3050 ENV 3,3,3,7,6,1,8,12,-3,4,7,3,3,3,-2,2
3060 ENT 1,4,-1,8,5,7,9,4,6,7,5,-5,9,15,9,8
3070 ENT 2,5,4,7,12,-1,9,6,1,8,8,-2,9,5,1,7
3080 ENT 3,4,-1,8,6,-2,9,12,3,5,12,-1,4,5,1,3
3090 SOUND 1,478,600,7,1,1,23
3100 SOUND 2,598,580,6,2,2,11
3110 SOUND 4,898,685,8,3,3,31
3120 GOTO 3400
3130 ' ***************************
3140 ' * RENCONTRE AVEC UNE ARME *
3150 ' ***************************
3160 VARM=INT(RND(1)*10)+2
3170 IF RIGHT$(ARM$(VARM),1)="e" THEN T$="e " ELSE T$=" "
3180  PRINT"Vous avez trouve un"T$:PRINT ARM$(VARM)" de valeur"VARM
3190 PRINT
3200 SOUND 1,478,24,5:SOUND 1,638,12,5:SOUND 1,478,12,5:SOUND 1,426,48,5
```

```
3210 SOUND 1,379,24,5:SOUND 1,478,12,5:SOUND 1,379,12,5:SOUND 1,319,48.5
3220 SOUND 1,284,12,5:SOUND 1,319,12,5:SOUND 1.284,12,5:SOUND 1,253,12,5
3230 SOUND 1,239,48,5
3240 '          *********************
3250 '          * ACHAT DE L'ARME ? *
3260 '          *********************
3270 IF FRC<VARM THEN PRINT"Vous ne pouvez pas achetercette arme":PRINT:G
OTO 750
3280 IF NbArmes>=12 THEN PRINT"Vous avez deja 12 armes":PRINT:GOTO 750
3290 INPUT"Voulez-vous l'acheter ?   (O/N) ";DESIR$
3300 IF DESIR$="O" OR DESIR$="o" THEN 3320
3310 IF DESIR$="N" OR DESIR$="n" THEN 750 ELSE 3290
3320 '         *******************************
3330 '         * CALCUL DE L'ACHAT DE L'ARME *
3340 '         *******************************
3350 ARM(PARAM)=VARM
3360 PARAM=PARAM+1
3370 NbArmes=NbArmes+1
3380 FRC=FRC-VARM
3390 GOTO 750
3400 ' *************************
3410 ' * COMBAT AVEC LE DRAGON *
3420 ' *************************
3430 PRINT"  Un monstrueux Dragon de force 12 se dresse devant vous ; sa
bouche crache   des Flammes.":PRINT:PRINT #5,"MONSTRE :":PRINT #5."Un Dra
gon de Force 12"
3440 INPUT"Voulez-vous le combattre ";REP$
3450 IF REP$="N" OR REP$="n" THEN 3900
3460 IF REP$="O" OR REP$="o" THEN 3480
3470 GOTO 3440
3480 '    *******************
3490 '    * CHOIX DE L'ARME *
3500 '    *******************
3510 PRINT:PRINT "Avec quelle arme voulez-  vous combattre ?":INPUT "  (
Si vous n'avez pas      d'arme tapez 0 )";VARM
3520 FOR V=0 TO 120
3530 IF ARM(V)=VARM OR VARM=0 THEN ABUS=0 : GOTO 3580
3540 NEXT V
3550 PRINT"Vous ne possedez pas cette arme ":ABUS=ABUS+1
3560 IF ABUS=10 THEN VARM=0:GOTO 3580
3570 GOTO 3480
```

```
3580 '   ********************
3590 '   * CALCUL DU COMBAT *
3600 '   ********************
3610 PRINT:PRINT"Pour engager le combat    tapez sur une touche":PRINT
3620 CA=CA+1
3630 IF INKEY$<>"" THEN 3650
3640 GOTO 3620
3650 VCOMB=12-VARM
3660 IF VARM>12 THEN VCOMB=0
3670 PRINT"Le combat est de force"VCOMB
3680 LOCATE #4,1,1:PRINT#4,"Des"
3690 FOR A=1 TO 20:SOUND 1,60,1,2
3700 DES=INT(RND(1)*10)+2
3710 DES$=(RIGHT$(STR$(DES),2))
3720 LOCATE #4,1,2:PRINT #4,DES$
3730 FOR B=1 TO 100:NEXT B
3740 NEXT A:SOUND 1,119,5,2
3750 IF DES<VCOMB THEN 3900
3760 IF DES>VCOMB THEN 3830
3770 IF DES=VCOMB THEN 3780
3780 '       ***********
3790 '       * EX EQUO *
3800 '       ***********
3810 PRINT" Vous n'avez pas reussi a terrasser ce Dragon car  les des on
t fait";DES;"Vous allez de nouveau lui etre confronte"
3820 GOTO 3440
3830 '       *********
3840 '       * GAGNE *
3850 '       *********
3860 DRAGON=1:CLS #5:FRC=FRC+VCOMB:LOCATE #1,1,2:PRINT #1,"    "FRC
3870 PRINT"Vous etes sorti victorieuxde ce terrible combat "
3880 FOR B=1 TO 2000:NEXT B
3890 CLS:IF ROUTI=6 THEN 4090 ELSE 2100
3900 '       ********
3910 '       * MORT *
3920 '       ********
3930 CLS:PRINT" Helas, les monstres du  Manoir de Rochebrune  ont eu rai
son de votre courageet de votre temerite, vousempechant ainsi de mener  a
 bien votre entreprise."
3940 PRINT:PRINT" Vous n'avez pas reussi a tuer le Pere de la Belle  Hele
ne prisonniere de la  Tour Noire, et a reunir   les 5 Tresors du Manoir."
3950 PRINT:PRINT" D'autres Chevaliers vont essayer de delivrer la   Prin
```

```
cesse mais votre prou-esse restera celebre dans les villaqes de ce triste
 pays.":PRINT
3960 GOSUB 4050
3970 SOUND 1,253,24,5:SOUND 1,284,24,5:SOUND 1,253,120,5
3980 GOSUB 4050
3990 SOUND 1,253,24,5:SOUND 1,239,24,5:SOUND 1,213,120,5:SOUND 1,190,24,5
4000 SOUND 1,179,48,5:SOUND 1,190,24,5:SOUND 1,213,48,5:SOUND 1,239,24,5
4010 SOUND 1,253,120,5
4020 GOSUB 4050
4030 SOUND 1,239,24,5:SOUND 1,253,24,5:SOUND 1,284,96,5
4040 END
4050 SOUND 1,284,24,5:SOUND 1,239,48,5:SOUND 1,284,24,5:SOUND 1,379,48,5
4060 SOUND 1,284,24,5:SOUND 1,239,120,5:SOUND 1,213,24,5:SOUND 1,190,24,5
4070 SOUND 1,213,24,5:SOUND 1,239,24,5:SOUND 1,253,24,5
4080 RETURN
4090 '   **********
4100 '   * GLOIRE *
4110 '   **********
4120 CLS:PRINT" Vous avez vaincu le Baronde Rochebrune et reuni les5 Tres
ors, ce qui vous   donne l'acces a la Tour   Noire dans laquelle vous  a
ttend la Belle Helene.":PRINT
4130 IF INKEY$="" THEN 4130 ELSE CLS
4140 PRINT" La Mort du Dragon marque la fin du malefice que le Baron avai
t jete sur ce   pays :":PRINT
4150 PRINT"Un vent violent chasse lesnuages noirs qui s'etaientamonceles
et decouvre un  ciel d'azur.":PRINT
4160 PRINT" Apres votre depart, les  villageois virent le Cha- teau s'emb
raser et quand  la fumee se dissipa, il nerestait plus pierre sur   pierr
e. Pendant des siecles rien ne poussa en ces   lieux."
4170 PRINT
4180 ENV 1,15,-1,5
4190 SOUND 1,478,32,15,1:SOUND 1,358,32,15,1
4200 SOUND 1,358,16,15,1:SOUND 1,319,16,15,1
4210 SOUND 1,284,32,15,1:SOUND 1,358,32,15,1
4220 SOUND 1,239,64,15,1:SOUND 1,284,48,15,1
4230 SOUND 1,284,16,15,1:SOUND 1,268,32,15,1
4240 SOUND 1,239,16,15,1:SOUND 1,268,16,15,1
4250 SOUND 1,284,16,15,1:SOUND 1,268,16,15,1
4260 SOUND 1,239,32,15,1:SOUND 1,319,16,15,1
4270 SOUND 1,358,16,15,1:SOUND 1,319,16,15,1
4280 SOUND 1,284,16,15,1:SOUND 1,319,32,15,1
4290 SOUND 1,478,32,15,1:SOUND 1,358,32,15,1
```

```
4300 SOUND 1,358,16,15,1:SOUND 1,319,16,15,1
4310 SOUND 1,284,32,15,1:SOUND 1,358,32,15,1
4320 SOUND 1,239,64,15,1:SOUND 1,284,48,15,1
4330 SOUND 1,284,16,15,1:SOUND 1,268,16,15,1
4340 SOUND 1,239,16,15,1:SOUND 1,284,16,15,1
4350 SOUND 1,268,16,15,1:SOUND 1,319,48,15,1
4360 SOUND 1,358,16,15,1:SOUND 1,358,96,15,1
```



1

LABELMATIC

Michel ARCHAMBAULT

Le rôle spécifique de ce programme de "qualité commerciale" est d'imprimer des adresses sur des étiquettes adhésives en continu, pour envois en nombre ("MAILING"). Il possède toutes les fonctions d'un gestionnaire de fichiers classique, à savoir : mise à jour, recherche, tri, etc. Le fait qu'il soit 100 % Basic ne nuit pas à la rapidité qui demeure plus que suffisante.

Il dispose en outre des caractères AZERTY accentués et de sécurités anti-plantage afin d'être confié à une personne sans formation informatique (vérifié).

Côté imprimante, il faut un modèle 80 colonnes de marque quelconque chargé avec des étiquettes en continu. Deux étiquettes par rangée de format normalisé 89×36, 102×36 ou 107×36 mm. Ce format importe peu car le centrage de chaque rubrique est automatique. Pour fixer les idées, disons le prix de revient moyen d'une étiquette adhésive est de l'ordre de 5 à 8 centimes TTC.

LE PROGRAMME

Ne sursautez pas à la vue de certains caractères bizarres au sein de certains mots ; ils donneront à l'écran et sur imprimante des caractères accentués : a commercial = à ; anti-slash = ç ; crochet ouvert + SHIFT (accolade ouverte) = é ; accolade fermée = è ; barre verticale = ù ; crochet ouvert = ° ; crochet fermé = § (définitions en GOSUB 51000).

LE MENU PRINCIPAL (1000)

Il propose dix options. Après l'exécution de chacune d'entre elles, il y a toujours retour à ce menu.

Chaque option débute par une confirmation (O/N) pour les étourdis. Le nombre de fiches en RAM (maxi ≃200) apparaît en haut à droite de l'écran.
Tous les "questionnaires" utilisent le module 50000. Exemple, la ligne 1150 : TEX$ rassemble tous les symboles d'options. Le module 50000 fait quatre opérations :
— il affiche, en bas de l'écran, la liste des réponses possibles,
— il transforme les réponses en majuscules,
— si la réponse n'est pas dans TEX$, il émet un BIP sonore et repose la question,
— il retourne au programme le numéro K de l'option choisie.

NOTE : Nous vous conseillons d'enregistrer d'abord sur une cassette à part les modules 50000 et 51000. Ils vous seront très utiles pour d'autres programmes ; il suffit de les joindre au programme en cours par MERGE" ".

LE FICHIER D'ADRESSES

Il est mis dans un tableau DIM AD$(200,5). Cela nous procure une grande souplesse d'utilisation (modification de fiches, recherche, tri, etc.). Seul défaut, ce grand tableau en RAM ralentit les opérations de sauvegarde sur cassette. Les "rubriques" ou "champs" de 0 à 5 sont les suivantes :
0 = nom, exemple "Mr. Jean BERGER" (22 caractères)
1 = adresse 1, exemple "6 rue MACHIN" (22 caractères)
2 = adresse 2, lieu-dit, etc. (22 caractères)
3 = code postal (5 caractères)
4 = ville (16 caractères)
5 = (facultatif) secteur de distribution (3 caractères) mais peut être porté à 11 caractères pour le numéro de téléphone, pays étranger ou toute autre remarque.
Nos longueurs maximales en caractères sont arbitraires (= minimum). Vous pouvez élargir.

LA PAGE D'ECRAN

Elle représente une seule fiche à la fois. Elle spécifie si vous êtes en saisie ou en vision, son numéro, le titre de chaque champ, sa longueur en pointillés (en saisie), et en bas, un menu de trois lettres.
S = fiche suivante
P = fiche précédente
F = FIN (retour au menu).
En saisie, une quatrième option "E" (Erreur), qui efface la fiche à l'écran.

NOTE : Grâce à la commande LINE INPUT au lieu de INPUT, on peut entrer des virgules et des guillemets.

LA VISION

La vision du fichier fiche par fiche peut se faire en donnant le numéro de fiche ou par recherche du nom, même *une partie* du nom. Ainsi, ''RCHAMB'' présentera la fiche ''ARCHAMBAULT'' ; ''RAT'' peut présenter la fiche de M. GARRAT, puis celle de Mme. RAT, mais '' RAT'' (un espace devant) ne conduira qu'à Mme. RAT.

LE TRI

Ou classement peut se faire d'après n'importe quel champ (classement alphabétique, ordre ASCII). Nous utilisons ici le ''tri à bulle'', la ''bulle'' étant l'indice zéro du tableau DIM. Un tri sur un grand désordre peut demander près de cinq minutes, mais s'il s'agit de mettre en place quelques fiches nouvellement entrées (''suite de saisie''), cela ne prend que quelques secondes.

LA SAUVEGARDE DU FICHIER

Elle débute par le nombre de fiches en RAM (NF), le nom du fichier (à entrer au clavier) et enfin le contenu des fiches. A la lecture, il n'est pas utile d'entrer ce nom (tapez ENTER).

L'IMPRESSION

Elle peut se faire sous deux modes. La ''GLOBALE'' ou ligne à ligne : sur une ligne, on a le numéro puis le contenu de la fiche. C'est un listing personnel. Enfin, le ''LABEL'' (=étiquette) qui vous demande le numéro de départ et le numéro final. Pour éditer tout le fichier, tapez ''1'' puis ''D'' (=dernier).

TRES IMPORTANT : En lignes 8100 et 9060, figure l'ordre CHR$(27)''R''CHR$(1) qui, sur les imprimantes de la marque EPSON (et assimilées), conduit aux caractères français (à, é, è, ù, ç,...). Consultez la notice de la vôtre afin de modifier ou confirmer ce code. Si votre imprimante n'a pas cette possibilité (par code ou par switches), supprimez cette commande.
Le listing présenté fait 7391 octets, soit 4 ''blocks''.

CONSEILS D'UTILISATION

Suppression de fiche
Appelez cette fiche par l'option modification ; elle apparaît vide. Mettez des "zzzzz" dans chaque champ. Si vous refaites un tri, elle ira en bout de liste. Sinon, laissez-la en place, mais lors de la prochaine suite de saisie, recherchez par "zzz", modification, et "remplissage" par la nouvelle personne.

Adresses commerciales
Utilisez le champ adresse 1 pour le nom du service, et "adresse 2" pour l'adresse proprement dite.

COMMANDE DU MAGNETO-CASSETTE

Il est normal que l'écran paraisse *bloqué* quelques instants (absence du curseur) *avant et après* un enregistrement de fichier. C'est l'AMSTRAD qui réorganise sa RAM, et le tableau DIM le gène un peu. Ce temps augmente avec la taille du fichier.
Un bug de l'AMSTRAD : au-delà de 90 fiches (environ), l'écran devrait afficher "Saving - nom du fichier - Block 1". Or, à la place du nom attendu, apparaissent parfois des fragments de chaînes du fichier ! Sans importance, le nom réel que vous avez tapé (= FICH$) est une des données enregistrées.

LA PLACE MEMOIRE

Avec notre fichier personnel de 112 fiches en RAM, il reste 25574 octets disponibles ! On doit donc pouvoir dépasser les 200 fiches initialement prévues. Toutefois, il est plus pratique de fractionner en fichiers de tailles raisonnables.
Pour savoir ce qui vous reste, tapez "F" au menu principal, puis PRINT FRE (" "), et enfin GOTO 1000 pour revenir au programme.

```
 1-Mr DUPONT Jacques:15 rue AMPERE La VIGERIE 47123 St MARTIN *Secteur 12
 2-Mlle HEVRARD Renée:8 Allée des LILAS  66234 MONTLOUIS *Secteur 15
 3-Mr,Mme GALVIN Paul:Villa "LES CHENES" Lotissement LE BOIS 77853 TRAMIENS
*Secteur 8
 4-DUCHNOCK SARL:Service du Personnel 12-56 Bd MAGENTA 21897 SAINT FAISANT
*Secteur 7
-----------------------------------------------------------------------------

        Mr DUPONT Jacques                    Mlle HEVRARD Renée
          15 rue AMPERE                       8 Allée des LILAS
           La VIGERIE
         47123 St MARTIN                       66234 MONTLOUIS
               12                                    15

       Mr,Mme GALVIN Paul                      DUCHNOCK SARL
       Villa "LES CHENES"                   Service du Personnel
       Lotissement LE BOIS                    12-56 Bd MAGENTA
         77853 TRAMIENS                     21897 SAINT FAISANT
               8                                     7
```

```
10 'LABEL-MATIC - Amstrad - M.A. 1985
100 MODE 0:BORDER 16:INK 0,1:INK 1,24:PAPER 11:CLS
110 LOCATE 1,7:PEN 12:PRINT "L A B E L--M A T I C"
120 LOCATE 1,13:PEN 4:PRINT "Editeur d'Etiquettes
130 LOCATE 2,21:PEN 0:PRINT "M.Achambault-1985"
140 FOR I=1 TO 3000:NEXT
200 MODE 1:BORDER 2:PAPER 0:PEN 1:CLS
210 DIM AD$(200,5):DEFINT A-Z
300 GOSUB 51000
1000 'MENU PRINCIPAL
1010 CLS:INK 0,1:INK 1,24
1020 LOCATE 17,2:PEN 3:PRINT "M E N U":PEN 2:LOCATE 29,2:PRINT NF;"fiches
":PEN 1
1030 LOCATE 6,4:PRINT "N = Nouveau fichier"
1040 LOCATE 6,6:PRINT "C = Chargement-lecture d'un fichier"
1050 LOCATE 6,8:PRINT "S = Suite d'un fichier charg{"
1060 LOCATE 6,10:PRINT "M = Modification du fichier"
1070 LOCATE 6,12:PRINT "E = Enregistrement sur cassette"
1080 LOCATE 6,14:PRINT "V = Visionner ou rechercher"
1090 LOCATE 6,16:PRINT "L = impression en LABELS"
1100 LOCATE 6,18:PRINT "G = impression GLOBALE(en lignes)"
1105 LOCATE 6,20:PRINT "T = TRI du Fichier"
1110 LOCATE 6,22:PRINT "F = Fin"
1150 TEX$="NCSMEVLGTF":GOSUB 50000
1160 ON K GOTO 2000,3000,4000,5000,6000,7000,8000,9000,12000,1500
1170 GOTO 1000
1500 'FINAL
1510 CLS:LOCATE 10,12:PRINT "VOUS POUVEZ ETEINDRE.":PRINT:PRINT
1520 END
2000 ' NOUVEAU FICHIER
2010 CLS
2020 LOCATE 6,4:PRINT "CREATION D'UN NOUVEAU FICHIER:"
2030 LOCATE 12,7:PRINT "( 200 fiches maxi )"
2040 LOCATE 3,12:PRINT "Nous effa\ons le Fichier en M{moire."
2050 LOCATE 15,14:PRINT "D'ACCORD ?"
2060 TEX$="ON":GOSUB 50000
2070 IF K=2 THEN 1000
2080 N=1:NF=0:GOSUB 10000
2090 NF=N:GOTO 1000
3000 ' CHARGEMENT FICHIER
3010 CLS:LOCATE 3,3:PRINT "LECTURE D'UN FICHIER SUR CASSETTE:"
3020 LOCATE 16,10:PRINT "D'ACCORD ?"
```

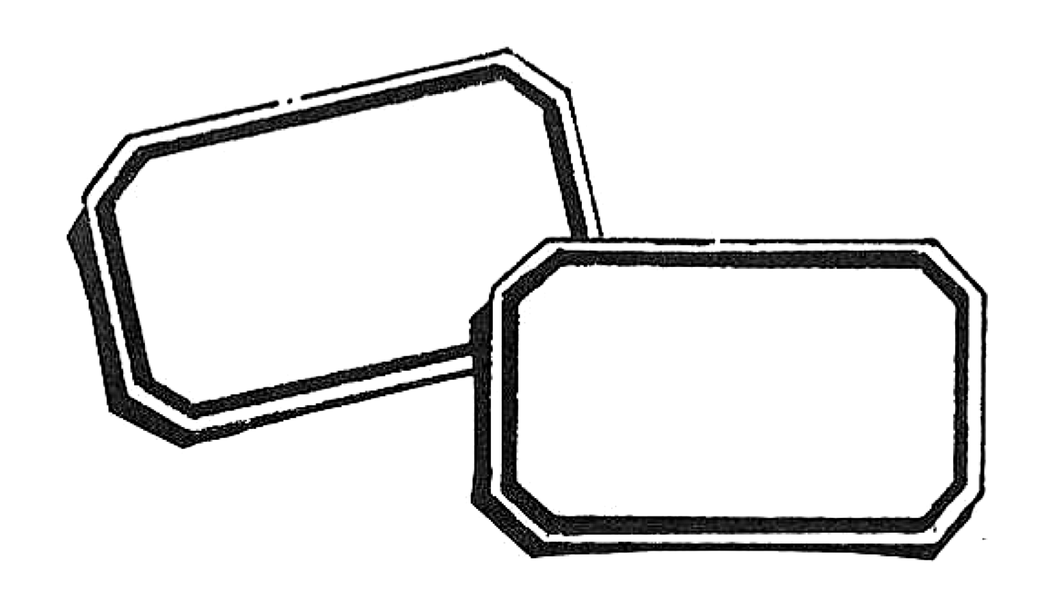
```
3030 TEX$="ON":GOSUB 50000
3040 IF K=2 THEN 1000
3050 CLS:LOCATE 12,5:LINE INPUT "NOM DU FICHIER: ",FICH$
3060 LOCATE 3,8:PRINT "D{s que la cassette est positionn{e"
3070 LOCATE 11,10:PRINT "Pressez une Touche."
3080 WHILE INKEY$="":WEND
3090 OPENIN FICH$
3100 INPUT #9,NF,FICH$:FOR I=1 TO NF:FOR J=0 TO 5:INPUT #9,AD$(I,J):NEXT:
NEXT
3110 CLOSEIN
3120 LOCATE 12,15:PRINT "Il-y-a";nf;"Fiches."
3130 FOR I=1 TO 2500:NEXT
3140 GOTO 1000
4000 'SUITE DE SAISIE
4010 CLS
4020 LOCATE 2,4:PRINT "SUITE DE SAISIE DU FICHIER EN MEMOIRE."
4030 LOCATE 16,10:PRINT "D'ACCORD ?":TEX$="ON":GOSUB 50000
4040 IF K=2 THEN 1000
4050 N=NF+1:GOSUB 10000:NF=N
4060 GOTO 1000
5000 'MODIFICATION
5010 CLS
5020 LOCATE 10,5:PRINT "MODIFICATION DE FICHE:"
5030 LOCATE 12,11:PRINT "( Pour FIN --> 0 )"
5040 LOCATE 12,9:INPUT "QUEL NUMERO ?",NR$
5050 IF NR$="0" THEN 1000
5060 NR=VAL(NR$):IF NR=0 OR NR>NF THEN PRINT CHR$(7):GOTO 5040
5070 N=NR:GOSUB 10000:GOTO 5000
6000 'ENREGISTREMENT
6010 CLS:LOCATE 7,5:PRINT "ENREGISTREMENT SUR CASSETTE:"
6020 LOCATE 16,10:PRINT "D'ACCORD ?":TEX$="ON":GOSUB 50000
6030 IF K=2 THEN 1000
6040 LOCATE 1,24:PRINT CHR$(18):LOCATE 3,10:PRINT "Pr{parez une Cassette
vierge,avec"
6050 LOCATE 3,12:PRINT "amorce avanc{e."
6070 LOCATE 12,15:LINE INPUT"NOM DU FICHIER: ",FICH$
6080 SPEED WRITE 1:OPENOUT FICH$
6090 WRITE #9,NF,FICH$:FOR I=1 TO NF:FOR J=0 TO 5:WRITE #9,AD$(I,J):NEXT:
NEXT
6100 CLOSEOUT
6110 GOTO 1000
7000 'VISIONNER,RECHERCHER
```

```
7010 CLS:LOCATE 12,4:PRINT "VISION ou RECHERCHE:"
7020 LOCATE 6,8:PRINT "T = TOUS,depuis le d{but."
7030 LOCATE 6,10:PRINT "N = Num{ro."
7035 LOCATE 6,12:PRINT "R = Recherche d'un Nom"
7040 LOCATE 6,14:PRINT "F = FIN."
7050 TEX$="TNRF":GOSUB 50000
7060 ON K GOTO 7100,7200,7300,7500
7100 N=1:GOSUB 11000:GOTO 7000
7200 LOCATE 12,15:INPUT "QUEL NUMERO ?",NR$
7210 NR=VAL(NR$):IF NR=0 OR NR>NF THEN PRINT CHR$(7):GOTO 7200
7220 N=NR:GOSUB 11000:GOTO 7000
7300 LOCATE 12,17:LINE INPUT "NOM cherch{:",NM$
7310 FOR N=1 TO NF
7320 IF INSTR(3,AD$(N,0),NM$)>0 THEN GOSUB 11000
7330 NEXT
7500 GOTO 1000
8000 ' IMPRESSION 2 LABELS / 80 col.
8010 CLS:LOCATE 8,5:PRINT "IMPRESSION DES ETIQUETTES:":LOCATE 16,11:PRINT
 "D'ACCORD ?"
8020 TEX$="ON":GOSUB 50000:IF K=2 THEN 1000
8030 CLS:LOCATE 7,5:PRINT "Deux Etiquettes par Rang{e."
8040 LOCATE 12,8:INPUT "NUMERO DEPART: ",ND$:ND=VAL(ND$):IF ND=0 OR ND>=N
F THEN PRINT CHR$(7):GOTO 8040
8050 LOCATE 14,12:PRINT "( D pour dernier )"
8060 LOCATE 13,10:INPUT "NUMERO FINAL: ",NA$:IF UPPER$(NA$)="D" THEN NA=N
F:GOTO 8080
8070 NA=VAL(NA$):IF NA<=ND OR NA>NF THEN 8060
8080 LOCATE 10,17:PRINT "V{rifiez l'Imprimante,":LOCATE 9,19:PRINT "puis
pressez une Touche."
8090 WHILE INKEY$="":WEND
8100 PRINT #8,CHR$(27)"R"CHR$(1);
8120 FOR I=ND TO NA STEP 2
8130 PRINT #8,SPC((37-LEN(AD$(I,0)))/2);AD$(I,0);SPC((43-LEN(AD$(I,0)))/2
);SPC((40-LEN(AD$(I+1,0)))/2);AD$(I+1,0)
8140 PRINT #8,SPC((37-LEN(AD$(I,1)))/2);AD$(I,1);SPC((43-LEN(AD$(I,1)))/2
);SPC((40-LEN(AD$(I+1,1)))/2);AD$(I+1,1)
8150 PRINT #8,SPC((37-LEN(AD$(I,2)))/2);AD$(I,2);SPC((43-LEN(AD$(I,2)))/2
);SPC((40-LEN(AD$(I+1,2)))/2);AD$(I+1,2)
8160 PRINT #8,SPC((36-LEN(AD$(I,3))-LEN(AD$(I,4)))/2);AD$(I,3);" ";AD$(I,
4);SPC((42-LEN(AD$(I,3))-LEN(AD$(I,4)))/2);SPC((41-LEN(AD$(I+1,3))-LEN(AD
$(I+1,4)))/2);AD$(I+1,3);" ";AD$(I+1,4)
```

```
8170 PRINT #8,SPC(17);AD$(I,5);SPC(42);AD$(I+1,5)
8180 FOR J=1 TO 4:PRINT #8:NEXT
8190 NEXT
8200 GOTO 1000
9000 ' IMPRESSION GLOBALE
9010 CLS:LOCATE 12,5:PRINT "IMPRESSION GLOBALE:"
9020 LOCATE 12,7:PRINT "( ligne @ ligne )":LOCATE 16,11:PRINT "D'ACCORD ?
"
9030 TEX$="ON":GOSUB 50000:IF K=2 THEN 1000
9040 CLS:LOCATE 10,5:PRINT "Pr{parez l'Imprimante,":LOCATE 9,10:PRINT "pu
is pressez une Touche."
9050 WHILE INKEY$="":WEND
9060 PRINT #8,CHR$(27)"R"CHR$(1);SPC((80-LEN(FICH$))/2);FICH$
9070 PRINT #8:FOR I=1 TO NF
9090 PRINT #8,USING "###";I;:PRINT#8,"-";AD$(I,0);":";AD$(I,1);" ";AD$(I,
2);" ";AD$(I,3);" ";AD$(I,4);" *Secteur ";AD$(I,5):NEXT
9100 PRINT #8,STRING$(78,"_")
9110 GOTO 1000
10000 ' SAISIE
10010 CLS:INK 0,13:INK 1,0
10020 LOCATE 10,2:PRINT "SAISIE FICHE NUMERO";N
10030 LOCATE 13,5:PRINT STRING$(22,CHR$(144));CHR$(151)
10040 LOCATE 8,5:LINE INPUT "NOM: ",AD$(N,0)
10050 IF LEN(AD$(N,0))>22 THEN AD$(N,0)=LEFT$(AD$(N,0),22):PRINT CHR$(7)
10060 LOCATE 13,7:PRINT STRING$(22,CHR$(144));CHR$(151)
10070 LOCATE 2,7:LINE INPUT "ADRESSE 1: ",AD$(N,1)
10080 IF LEN(AD$(N,1))>22 THEN AD$(N,1)=LEFT$(AD$(N,1),22):PRINT CHR$(7)
10090 LOCATE 13,9:PRINT STRING$(22,CHR$(144));CHR$(151)
10100 LOCATE 2,9:LINE INPUT "ADRESSE 2: ",AD$(N,2)
10110 IF LEN(AD$(N,2))>22 THEN AD$(N,2)=LEFT$(AD$(N,2),22):PRINT CHR$(7)
10120 LOCATE 13,11:PRINT STRING$(5,CHR$(144));CHR$(151)
10130 LOCATE 2,11:INPUT "CODE POST: ",AD$(N,3)
10140 LOCATE 13,13:PRINT STRING$(16,CHR$(144));CHR$(151)
10150 LOCATE 6,13:LINE INPUT "VILLE: ",AD$(N,4)
10160 IF LEN(AD$(N,4))>16 THEN AD$(N,4)=LEFT$(AD$(N,4),16):PRINT CHR$(7)
10170 LOCATE 13,15:PRINT STRING$(3,CHR$(144));CHR$(151)
10180 LOCATE 4,15:INPUT "SECTEUR: ",AD$(N,5)
10190 LOCATE 2,19:PRINT "SUITE (S) , ERREUR (E) , FIN (F) ?"
10200 TEX$="SEF":GOSUB 50000
10210 IF K=1 THEN N=N+1:GOTO 10000
10220 IF K=2 THEN 10000
10230 RETURN
```

```
11000 'AFFICHAGE
11010 CLS:INK 0,13:INK 1,0
11020 LOCATE 10,2:PRINT "VISION FICHE NUMERO";N
11030 LOCATE 8,5:PRINT "NOM: ";AD$(N,0)
11040 LOCATE 2,7:PRINT "ADRESSE 1: ";AD$(N,1)
11050 LOCATE 2,9:PRINT "ADRESSE 2: ";AD$(N,2)
11060 LOCATE 2,11:PRINT "CODE POST: ";AD$(N,3)
11070 LOCATE 6,13:PRINT "VILLE: ";AD$(N,4)
11080 LOCATE 4,15:PRINT "SECTEUR: ";AD$(N,5)
11090 LOCATE 8,19:PRINT "Pr{c}dent . Suivant . Fin"
11100 TEX$="PSF":GOSUB 50000:ON K GOTO 11200,11300,11400
11200 N=N-1:IF N=0 THEN PRINT CHR$(7):N=1
11210 GOTO 11010
11300 N=N+1:IF N>NF THEN PRINT CHR$(7):N=NF
11310 GOTO 11010
11400 RETURN
12000 'TRI
12010 CLS:LOCATE 13,4:PRINT "TRI DU FICHIER:":LOCATE 16,11:PRINT "D'ACCOR
D ?":TEX$="ON":GOSUB 50000:IF K=2 THEN 1000
12020 CLS:LOCATE 13,4:PRINT "CRITERE DU TRI:":LOCATE 9,8:PRINT "N = NOM"
12030 LOCATE 9,10:PRINT "A = ADRESSE 1":LOCATE 9,12:PRINT "L = Lieu-dit (
adresse 2)":LOCATE 9,14:PRINT "C = Code postal"
12040 LOCATE 9,16:PRINT "V = VILLE":LOCATE 9,18:PRINT "S = SECTEUR"
12050 TEX$="NALCVS":GOSUB 50000:T=K-1:LOCATE 16,21:PRINT "Patience..."
12060 F=0:FOR J=1 TO NF
12070 IF AD$(J,T)>=AD$(J-1,T) THEN 12090
12080 FOR R=0 TO 5:AD$(0,R)=AD$(J,R):AD$(J,R)=AD$(J-1,R):AD$(J-1,R)=AD$(0
,R):NEXT:F=1
12090 NEXT
12100 IF F=1 THEN 12060
12110 GOTO 1000
50000 'REPONSE A UN MENU
50010 LT=LEN(TEX$):R$=""
50020 LOCATE 15-LT,24:PRINT "R{ponse (";
50030 FOR I=1 TO LT-1
50040 PRINT MID$(TEX$,I,1);",";:NEXT
50050 PRINT RIGHT$(TEX$,1);")";CHR$(154);CHR$(243);CHR$(207)
50060 TEX$=UPPER$(TEX$)
50070 WHILE R$="":R$=INKEY$:WEND
50080 R$=UPPER$(R$):K=INSTR(TEX$,R$)
50090 IF K=0 THEN R$="":PRINT CHR$(7);:GOTO 50070
```

```
50100 RETURN
51000 ' AZERTY ACCENTUE
51010 SYMBOL AFTER 7
51020 SYMBOL 64,96,48,120,12,124,204,118,0
51030 SYMBOL 91,0,56,108,56,0,0,0,0
51040 SYMBOL 92,0,0,60,102,96,102,60,24
51050 SYMBOL 93,60,96,60,102,60,6,60,0
51060 SYMBOL 123,12,24,60,102,126,96,60,0
51070 SYMBOL 124,48,24,102,102,102,102,62,0
51080 SYMBOL 125,48,24,60,102,126,96,60,0
51100 RETURN
65535 'FIN DE LISTING
```



1

LA MÉMOIRE D'ÉCRAN

Michel ARCHAMBAULT

Les notions exposées dans cet article ne sont pas du tout indispensables pour devenir un bon progammeur en Basic ; elles permettent d'aller plus loin en triturant la zone mémoire de l'écran par des PEEK ou des POKE. Pour faire quoi ?

Tout d'abord faire des programmes de "HARD COPY" qui exécutent sur imprimante une sorte de photocopie de l'écran en cours, surtout utile pour les graphismes (vous pourrez utiliser ces programmes même si vous ne comprenez pas leurs fonctionnements). Avec ces notions, et de la patience, vous pourrez aussi créer des effets colorés inaccessibles par les seules fonctions Basic.

GENERALITES SUR LA MEMOIRE

Le CPC 464, comme la plupart des micro-ordinateurs de moins de 20 000 FF, utilise un microprocesseur "8 bits" (un octet). Pour aller chercher un octet en mémoire, il faut que ce dernier ait une adresse. Un octet peut prendre 256 valeurs possibles ($256 = 2^8$). Pour les adresses, disons un octet pour les "colonnes" et un octet pour les "lignes" d'un "tableau d'adresses", donc $256 \times 256 = 65536$ "cases mémoire", qu'on numérote de 0 à 65535. C'est ce qu'on appelle 64 k-octets car le "kilo-octet" vaut en réalité 1024 octets... Donc 64 k-o est le maximum pour un micro 8 bits.
Il faut les partager entre la ROM (le Basic résident) et la RAM (programmes + variables en cours + les points lumineux envoyés à l'écran). Avec un Basic de 32 k-o, il ne resterait que 32 k-o pour la RAM, d'où vingt et quelques kilo-oktets disponibles pour programmes plus fichiers, c'est ce qui se passe avec les MSX et bien d'autres. Nous, nous avons 32 k-o de Basic, 43 k-o disponibles + 16 k-o réservés à l'écran ! La raison de ce "miracle" est que l'AMSTRAD "recharge et décharge"

la portion de la ROM dont il a besoin (c'est une vue d'esprit). De ce fait, une large partie de la RAM est plutôt ''mouvante'', ce qui va donner bien du plaisir aux amateurs de PEEK et de POKE...
La partie mémoire écran occupe une zone fixe, de 49152 à 65487, mais le fait que l'on puisse mélanger texte et graphisme, et ce en trois MODES, fait que sa structure est d'une extrême complexité inouie (même principe que sur les APPLE II, mais en pire). C'est ce qui explique que l'affichage des caractères est assez lent, exemple, quand on fait LIST sur un long programme.
Dans le manuel, les adresses sont indiquées en hexadécimal (préfixe & et quatre caractères). Il est pratique de savoir faire les conversions hexadécimal-décimal dans les deux sens (Note : le préfixe hexadécimal ne signifie pas ''six'' mais ''seize'').

LES CONVERSIONS HEXA/DECIMAL/BINAIRE

En binaire, on a deux symboles 1 et 0 ; en décimal, on a dix symboles 0 à 9 ; en hexa, on en a seize, 0 à 9 + A, B, C, D, E et F (&F=15;&A=10).
On représente la valeur d'un octet par deux symboles hexa, ainsi &FA=(256×15)+10= 250. Il suffit de taper PRINT &FA et 250 apparaît à l'écran. De même, &FF=255, c'est bien le maximum pour un octet.
Une adresse tient sur deux octets, exemple &AFC8. Vous ne pouvez pas faire PRINT &AFC8 car l'AMSTRAD ne sait pas faire la conversion sur plus d'un octet, vous êtes obligé de décomposer :

PRINT &AF*256+&C8

et la réponse apparaît : 45000.
L'opération inverse est beaucoup plus facile :

PRINT HEX$(45000)

l'écran répond AFC8 (le & est sous-entendu).
Pour traduire une longue suite de valeurs hexa en décimal, vous gagnerez du temps en utilisant ce très court programme :

```
10 '  CONVERSION HEXA --> DECIMAL
20 INPUT "&",H$
30 H$=RIGHT$("0000"+H$,4)
40 G$="&"+LEFT$(H$,2):D$="&"+RIGHT$(H$.2)
50 PRINT VAL(G$)*256+VAL(D$)
60 PRINT:GOTO 20
70 END
```

Passons à la représentation binaire d'un nombre. Tapez, par exemple : PRINT BIN$(45956), vous obtenez une suite de 16 bits 1 et 0. Essayez maintenant PRINT BIN$(3), vous obtenez "1" (c'est une chaîne) ; avec PRINT BIN$(3,16), vous avez 14 zéros suivis de "11".
Gare à l'opération inverse par &X, elle traduit cette "image binaire" en un nombre dit entier, c'est-à-dire compris entre −327869 et +32767 ! Un zéro suivi de quinze 1 donne 32767, tandis que 1 suivi de quinze zéros donne −32768.
Heureusement, vous n'avez pas à vous en préoccuper car dans la pratique les seules images binaires qui nous intéressent sont *sur un octet,* donc de 1 à 255.

LE PLAN MEMOIRE DE L'ECRAN

Nous y voilà !
Deux cent lignes horizontales de 80 octets chacune. Dans une ligne, les numéros d'octets se suivent de gauche à droite. Prenons le cas le plus simple, celui du MODE 2 : chaque bit représente un "pixel" ou point à l'écran, ainsi 255 fait un tiret continu (huit bits 1). La ligne en haut de l'écran va de 49152 à 49231, soit 80 × 8 = 640 points (ce qui explique qu'en coordonnées graphiques l'échelle horizontale aille de 1 à 640).
Hélas, l'octet suivant, le 49232, n'est pas au départ de la deuxième ligne mais huit lignes plus bas ! Et ainsi de suite au pas de 8 lignes. Après un premier "passage" de 25 lignes espacées de 8, commence le deuxième passage sur les lignes du dessous. Pour tout comprendre, essayez ceci :

```
100 ' BALAYAGE DE L'ECRAN
110 MODE 2
120 FOR A=49152 TO 65487
130 POKE A,3
140 NEXT
150 END
```

Un joli store vénitien !
Les figures 1 et 2 vous donnent le plan des adresses d'écran. Vous constatez qu'entre deux "tirets" (deux octets) situés à l'écran, l'un en-dessous de l'autre, il y a un écart de 2048 octets (= 2k-octets), qu'entre deux lignes d'un même "passage" (donc espacées de 8 lignes d'écran), il y a un écart de 14336 octets.
A présent, modifiez la ligne 110 en mettant MODE 1. Surprise ! Les traits sont rouges (=PEN 3). En ligne 110, mettez MODE 0 : les traits clignotent en rose/bleu ciel (=PEN 15). Nous abordons alors le code des couleurs en fonction du MODE.

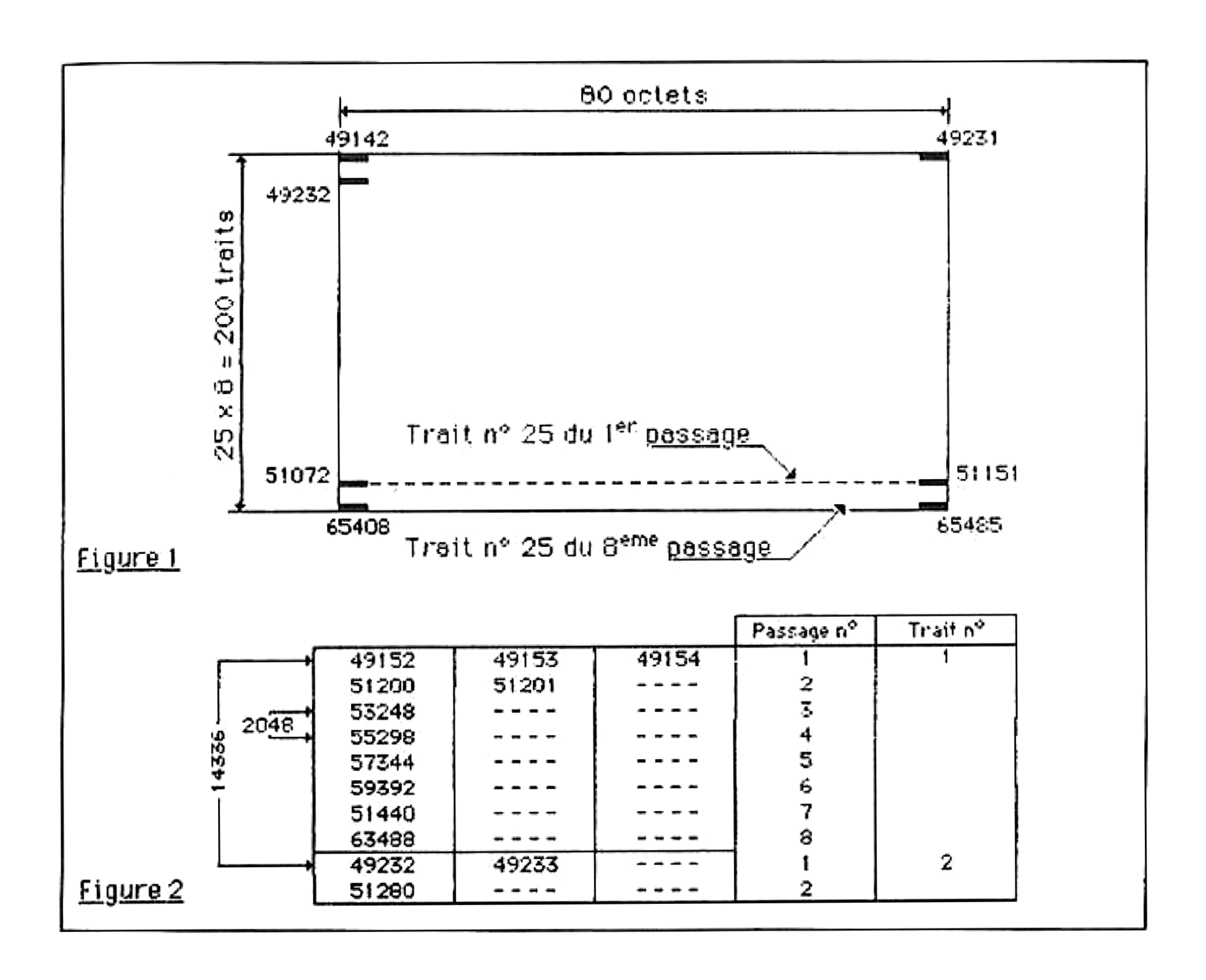


Figure 1

			Passage n°	Trait n°
49152	49153	49154	1	1
51200	51201	----	2	
53248	----	----	3	
55298	----	----	4	
57344	----	----	5	
59392	----	----	6	
51440	----	----	7	
63488	----	----	8	
49232	49233	----	1	2
51280	----	----	2	

Figure 2

LE CODAGE DES COULEURS

En MODE 2, rien de plus simple : comme il n'y a que deux couleurs possibles, un bit 1 fait un pixel PEN, un bit 0 un pixel PAPER (Pixel est la contraction anglaise de PICTURAL ELEMENT).
Lancez ce programme qui va dessiner en 0 et 1 les images binaires des quatre premiers caractères sur un écran en MODE 2.

```
200 '  IMAGE BINAIRE D'UN CARACTERE
210 MODE 2
220 CLS:PRINT "aV8/"
230 FOR A=49152 TO 63491 STEP 2048
240 FOR N=0 TO 3
250 CB=PEEK(A+N)
260 PRINT BIN$(CB,8);".";
270 NEXT:PRINT CHR$(13)
280 NEXT
290 END
```

Le résultat est spectaculaire. On y retrouve les compositions illustrées dans le manuel AMSTRAD page A3.2 et suivantes.
A présent, modifions la ligne 210 afin d'avoir MODE 1 (ou faites MODE 1:RUN 220). Oh ! Que c'est vilain ! Sur chaque "pavé", on n'a qu'une moitié de chaque caractère. Il faudrait accoller les moitiés gauches des pavés pour reconstituer le texte. Les moitiés droites des pavés (ou des octets) ne comportent que des 0. Ces zéros constituent en fait le code couleur de PEN (PEN 1 en ce cas) : apparions – les gauche + droite, on a '01" quand il y a quelque chose à tracer, or &X01 = 1 (PEN 1).
Tapons PEN 2 et, toujours en MODE 1, faisons RUN 220 : les 1 sont à droite et les zéros à gauche, &X10 = 2 (PEN 2).
Tapons PEN 3 et même manœuvre. Dans chaque pavé, le demi caractère est répété à gauche et à droite, or &X11 = 3.
Puisque dans chaque octet on associe un bit de la moitié droite à un de la moitié gauche, il n'y a que quatre combinaisons possibles :

"00" (=0) ; "01" (=1)
"10" (=2) et "11" (=3)

C'est pourquoi on n'a que quatre couleurs en MODE 1.
Et que se passe-t-il sur l'écran vidéo ? Sur chaque image binaire d'un octet (le "tiret"), chaque bit 1 est *doublé*, et dans la couleur du PEN. Un caractère occupe donc deux pavés, et comme il y a 80 octets par ligne, cela fait bien 40 caractères par ligne d'écran.
En MODE 0, c'est le même principe, mais en divisant par deux, un quart de caractère par pavé. Pour apparier les bits, on a alors 16 combinaisons possibles (les seize couleurs). A l'écran, chaque bit 1 est quadruplé dans la même couleur.

LE HARD COPY D'ECRAN

Il faut que votre imprimante possède l'instruction "BIT IMAGE", c'est le cas de la plupart des imprimantes "à aiguilles" ou "matricielles". En usage normal, une imprimante traduit l'octet reçu en dessinant le caractère ayant ce code ASCII, ici c'est beaucoup plus simpliste. Les huit aiguilles de la tête d'impression illustrent l'image binaire de cet octet. Ces dessins de bits vont être des "barres" verticales, or sur l'écran ils sont horizontaux ! Il suffit de tourner notre graphisme d'un quart de tour sur le papier. L'imprimante commencera par le bord vertical gauche en terminant par le bord droit de l'écran ; le bas de l'écran sera donc le long de la marge gauche du papier. Ce n'est pas gênant. L'auteur utilise une EPSON RX80 (les modèles FX ont les mêmes codes). Si la vôtre est d'une autre marque, il va falloir modifier les libellés des codes de commande ; nous légenderons les nôtres afin que vous puissiez les traduire pour votre machine. Si vous trouvez les mêmes,

ne soyez pas surpris car une bonne quinzaine de ''marques'' sont des EPSON rebaptisées.
Il y a deux programmes différents, pour le MODE 2 et le MODE 1 (la version MODE 0 ne satisfait pas l'auteur).
Il s'agit en fait de sous-programmes utilitaires

```
55000 'HARD COPY AMSTRAD MODE 1
55010 PRINT #8
55020 DIM Z(200)
55030  PRINT #8,CHR$(27);CHR$(64);CHR$(27);"3";CHR$(24);
55040 PRINT #8,CHR$(13)
55050 PRINT #8,CHR$(15);
55060 FOR COL=49152 TO 49231 STEP 2:I=0
55070 FOR LI=COL TO COL+1920 STEP 80
55080 FOR T=LI TO LI+14336 STEP 2048
55090 I=I+1:Z$=LEFT$(BIN$(PEEK(T),8),4)+   LEFT$(BIN$(PEEK(T+1),8),4)
55100 Z(I)=VAL("&X"+Z$):IF I=200 THEN GOSUB 55130
55110 NEXT:NEXT:PRINT #8,CHR$(13):NEXT
55120 RETURN
55130 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
55140 FOR J=200 TO 101 STEP -1
55150 PRINT #8,CHR$(Z(J));:NEXT
55160 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
55170 FOR J=100 TO 1 STEP -1
55180 PRINT #8,CHR$(Z(J));:NEXT
55190 RETURN
```

Temps d'exécution 4'15''. Format 112 × 84 mm.

```
56000 'HARD COPY MODE 2 AMSTRAD
56010 PRINT #8
56020 DIM Z(200)
56030  PRINT #8,CHR$(27);CHR$(64);CHR$(27);"A";CHR$(8);
56040 PRINT #8,CHR$(13)
56050 PRINT #8,CHR$(15);
56060 FOR COL=49152 TO 49231:I=0
56070 FOR LI=COL TO COL+1920 STEP 80
56080 FOR T=LI TO LI+14336 STEP 2048
56090 I=I+1:Z(I)=PEEK(T):IF I=200 THEN GOSUB 56120
56100 NEXT:NEXT:PRINT #8,CHR$(13):NEXT
56110 RETURN
```

```
56120 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
56130 FOR J=200 TO 151 STEP -1:P=Z(J)
56140 PRINT #8,CHR$(P);CHR$(0);:NEXT
56150 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
56160 FOR J=150 TO 101 STEP -1:P=Z(J)
56170 PRINT #8,CHR$(P);CHR$(0);:NEXT
56180 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
56190 FOR J=100 TO 51 STEP -1:P=Z(J)
56200 PRINT #8,CHR$(P);CHR$(0);:NEXT
56210 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
56220 FOR J=50 TO 1 STEP -1:P=Z(J)
56230 PRINT #8,CHR$(P);CHR$(0);:NEXT
56240 RETURN
```

Temps d'exécution 7'30''. Format 225 × 170 mm.

LEGENDES DES CODES EPSON

CHR$(27);CHR$(64) = vide le buffer (initialisation)
CHR$(27);''3'';CHR$(24) = interlignes de 24/216 de pouce
CHR$(27);''A'';CHR$(8) = interlignes de 8/72 de pouce
CHR$(15) = caractères condensés (17 par pouce)
CHR$(27);''K'' = mode bit-image

LE PRINCIPE

On explore des bandes verticales de l'écran de largeur 1 octet. Ces 200 octets sont mis en DIM Z puis ils sont retransmis à l'imprimante qui écrit donc une ligne. Passage à la colonne précédente et ainsi de suite. En jouant sur divers paramètres d'impression, nous sommes parvenus à restituer les proportions hauteur × largeur de l'écran (approximativement). En MODE 0, nous ne sommes pas (encore) parvenus à obtenir des dimensions d'image imprimées satisfaisantes, d'où l'absence de la version MODE 0.
Très important : en MODE 1, les inscriptions doivent être en PEN 1 ou en PEN 3 (pas en PEN 2) car nous ne prenons que la moitié gauche des images binaires (ligne 55090).
Suite à un petit bug de la ROM de l'EPSON RX 80, nous n'envoyons en ''bit image'' que des suites de 50 ou 100 octets à la fois.

REMARQUE

C'est aussi pour des raisons de proportions d'écran que les coordonnées graphiques sont de 640 × 400 alors que logiquement ce serait 640 × 200.

CODAGE BINAIRE D'UN CARACTERE

Vous voulez définir un caractère qui se trouve être un caractère existant mais légèrement modifié. Vous trouvez fastidieux de redessiner sa grille 8 × 8 et de totaliser toutes ces lignes. Alors, utilisez ce programme qui vous fait instantanément tout ce travail.

```
300 ' CODAGE BINAIRE D'UN CARACTERE
310 INPUT "CARACTERE OU CODE ASCII":C$:MODE 1
320 IF VAL(C$)>32 THEN PRINT CHR$(VAL(C$));SPC(2);C$:PRINT:GOTO 340
330 PRINT C$;SPC(2):ASC(C$):PRINT
340 FOR A=49152 TO 63488 STEP 2048
350 G$=LEFT$(BIN$(PEEK(A),8),4)
360 D$=LEFT$(BIN$(PEEK(A+1),8),4)
370 C$=G$+D$
380 PRINT C$;SPC(2);VAL("&X"+C$)
390 NEXT
400 PRINT:GOTO 310
410 END
```

Tapez indifféremment le caractère, s'il existe, au clavier ou son code ASCII. L'écran vous soumet sa fiche signalétique complète, à savoir : Le caractère, son code ASCII, sa configuration binaire en "1" et "0" et les totaux de chaque ligne. Exemple avec X majuscule : Code ASCII 88, codage 198, 108, 56, 56, 108, 198, 198, 0.
Faites un autre essai en tapant 251.
Ce programme refuse certains signes de ponctuation tels que + – , . &. En ce cas entrez le code ASCII.

1

UN DÉSASSEMBLEUR

Eddy DUTERTRE

Voici pour les amoureux du langage machine un petit désassembleur écrit en Basic qui leur permettra, entre autres, de percer les mystères du CPC 464. Dès le lancement par RUN, un choix est demandé par l'ordinateur quant au type de zone où doit être lancé le désassembleur (RAM ou ROM).

En effet, dans l'AMSTRAD, la ROM n'est pas accessible directement, elle se trouve en ''OVERLAY'' sur la RAM et, pour sélectionner l'une ou l'autre, une petite routine en langage machine est nécessaire. Cette sélection faite, il suffit ensuite de donner en *hexadécimal* l'adresse de lancement du désassembleur.
Dans le bas de l'écran, sont données quelques indications sur les options disponibles. La plus à gauche, intitulée ''SELECTION'' donne l'état RAM/ROM choisi précédemment. Les autres indications constituent le mode d'emploi. On y trouve :

ESPACE = Suite
Le désassemblage se fait par page d'écran, et pour poursuivre quand ce dernier est plein, il suffit d'appuyer sur la touche ''Espace''.

N = Nouvelle adresse
Après avoir désassemblé une page, il peut être nécessaire de sauter à une autre adresse. Ceci se fait en appuyant sur la touche ''N'' afin de revenir au menu.

I-Imprimante ON/OFF
Si l'on possède une imprimante, il peut être utile de conserver quelques listes désassemblées. Ceci se fait par appui sur la touche ''I'' constituant un interrupteur de marche-arrêt de l'imprimante. L'état est donné en permanence au bas de l'écran et au lancement du programme, l'imprimante est inhibée. Un appui sur ''I'' autorisera l'impression des prochaines lignes, un deuxième interdira, etc...

A = Arrêt
Sortie du programme et réinitialisation de l'écran.

(Les instructions sur IX et IY ne sont pas désassemblées).

```
10 REM *******************************
20 REM **    -DESASSEMBLEUR-        **
30 REM **                           **
40 REM **    @-DUTERTRE EDDY        **
50 REM **                           **
60 REM **      25/02/1985           **
70 REM **                           **
80 REM *******************************
90 DATA NOP,"2LD BC,","LD (BC),A",INC BC,INC B,DEC B,"1LD B,",RLCA,"EX AF
,AF'","ADD HL,BC"
100 DATA"LD A,(BC)",DEC BC,INC C,DEC C,"1LD C,",RRCA,"3DJNZ ","2LD DE,","
LD (DE),A",INC DE
110 DATA INC D,DEC D,"1LD D,",RLA,"3JR ","ADD HL,DE","LD A,(DE)",DEC DE,I
NC E,DEC E
120 DATA"1LD E,",RRA,"3JR NZ,","2LD HL,","4LD (    ),HL",INC HL,INC H,DEC
 H,"1LD H,",DAA
130 DATA"3JR Z,","ADD HL,HL","7LD HL,(    )",DEC HL,INC L,DEC L,"1LD L,",
CPL,"3JR NC,","2LD SP,"
140 DATA"4LD (    ),A",INC SP,INC (HL),DEC (HL),"1LD (HL),",SCF,"3JR C,",
"ADD HL,SP","5LD A,(    )",DEC SP
150 DATA INC A,DEC A,"1LD A,",CCF
160 DATA RET NZ,POP BC,"2JP NZ,","2JP ","2CALL NZ,",PUSH BC,"1ADD A,",RST
 0,RET Z,RET,"2JP Z,",,"2CALL Z,","2CALL ","1ADC A,",RST 8,RET NC,POP DE,
"2JP NC,","6OUT   ,A"
170 DATA"2CALL NC,",PUSH DE,"1SUB ",RST 16,RET C,EXX,"2JP C,","1IN A,","2
CALL C,",INST/IX,"1SBC A,",RST 24,RET PO,POP HL,"2JP PO,","EX (SP),HL","2
CALL PO,",PUSH HL,"1AND ",RST 32
180 DATA RET PE,JP (HL),"2JP PE,","EX DE,HL","2CALL PE,",-,"1XOR ",RST 40
,RET P,POP AF,"2JP P,",DI,"2CALL P,",PUSH AF,"1OR ",RST 48,RET M,"LD SP,H
L","2JP M,",EI
190 DATA "2CALL M,",INST/IY,"1CP ",RST 56
200 DATA LDI,CPI,INI,OUTI,,,,,LDD,CPD,IND,OUTD,,,,,LDIR,CPIR,INIR,OTIR,,,
,,LDDR,CPDR,INDR,OTDR
210 DATA"IN B,(C)","OUT (C),B","SBC HL,BC","4LD (    ),BC",NEG,RET N,IM 0
,"LD I,A","IN C,(C)","OUT (C),C","ADC HL,BC","7LD BC,(    )",,RETI,,"LD R
,A","IN D,(C)","OUT (C),D","SBC HL,DE","4LD (    ),DE"
220 REM ne pas effacer cette ligne
```

```
230 DATA B,C,D,E,H,L,(HL),A,B,C,D,E,H,L,(HL),A
240 DATA RLC,RRC,RL,RR,SLA,SRA,,SRL,BIT,RES,SET
250 DATA"ADD A,","ADC A,","SUB ","SBC A,","AND ","XOR ","OR ","CP "
260 DATA&DF,&04,&AB,&C9,&07,&AB,&FC,&3A,&00,&00,&32,&0E,&AB,&C9,&00,&00,&
CD,&60,&BB,&32,&17,&AB,&C9
270 MEMORY  &9FFF: DIM R$(16),S$(11)
280 RESTORE 230:Q=0
290 FOR N=0 TO 15:READ R$(N):NEXT
300 FOR N=0 TO 10:READ S$(N):NEXT
310 FOR N=0 TO 7:READ T$(N):NEXT
320 FOR N=&AB00 TO &AB16:READ A:POKE N,A:NEXT
330 MODE 1:WINDOW 1,40,1,20:WINDOW #1,1,40,21,25:ORIGIN 0,64:DRAW 639,0,1

340 ORIGIN 0,46:DRAW 639,0,1:ORIGIN 144,0:DRAW 0,64,1:LOCATE#1,1,2:PRINT#
1,"SELECTION"
350 LOCATE #1,19,2:PRINT#1,"COMMANDES":LOCATE #1,11,4:PRINT#1,"ESPACE=Sui
te   N=Nouv.adresse"
360 LOCATE #1,11,5:PRINT#1,"I=Imprimante ON/";CHR$(24);"OFF";CHR$(24)
370 LOCATE #1,33,5:PRINT#1,"A=Arret"
380 CLS: PRINT"VOUS VOULEZ DESASSEMBLER UNE ZONE DE:"
390 PRINT:PRINT"1-RAM":PRINT"2-ROM":PRINT:PRINT"VOTRE CHOIX ?"
400 G$=INKEY$:G$=UPPER$(G$):IF G$="" OR (G$<>"1" AND G$<>"2" AND G$<>"I")
 THEN 400
410 IF G$="I" THEN GOSUB 1340:GOTO 400
420 PRINT:PRINT:R=VAL(G$):LOCATE #1,4,4:IF R=1 THEN PRINT#1,"RAM" ELSE PR
INT#1,"ROM"
430 A=0:B$="":PRINT "ADRESSE DE DESASSEMBLAGE:...."
440 FOR N=0 TO 3
450 A$=INKEY$:a$=UPPER$(a$):IF A$="" THEN 450
460 IF A$="I" THEN GOSUB 1340:GOTO 450
470 IF A$>"9" AND A$<"A" THEN 450
480 IF A$>"F" OR A$<"0" THEN 450
490 LOCATE 26+N,9:PRINT A$:B$=B$+A$:NEXT
500 A=VAL("&"+B$):IF A<0 THEN A=65536+A
510 CLS:FOR N=1 TO 20
520 PRINT UPPER$(B$);"     ";
530 P=A:GOSUB 1290:C$=HEX$(D,2):GOSUB 670:GOSUB 680
540 LOCATE 27,N:FOR M=A TO A+I-1:P=M:GOSUB 1290:PRINT HEX$(D,2);" ";:NEXT
:PRINT
550 A=A+I:IF A>65535 THEN A=0
560 B$=HEX$(A,4):IF Q=0 THEN 580
570 FOR M=1 TO 40:LOCATE M,N:CALL &AB10:PRINT#8,CHR$(PEEK(&AB17));:NEXT M
```

```
:PRINT#8
580 NEXT N
590 T$=INKEY$:T$=UPPER$(T$):IF T$="" THEN 590
600 IF T$=" " THEN B$=HEX$(A,4):GOTO 510
610 IF T$="I" THEN GOSUB 1340:GOTO 590
620 IF T$="N" THEN 380
630 IF T$="A"THEN WINDOW 1,40,1,25:CLS:END
640 GOTO 590
650 DATA IM 1,"LD A,I","IN E,(C)","OUT (C),E","ADC HL,DE"."7LD DE.(     )"
,,,IM 2,"LD A,R","IN H,(C)","OUT (C),H","SBC HL,HL","4LD (     ),HL",,,,RR
D,"IN L,(C)","OUT (C),L","ADC HL,HL","7LD DE,(     )",,,,RLD,,,"SBC HL,SP"
660 DATA "4LD (     ),SP",,,,,"IN A,(C)","OUT (C),A","ADC HL,SP","7LD SP,(
    )"
670 Y=VAL("&"+MID$(C$,1,1)):Z=VAL("&"+MID$(C$,2,1)):RETURN
680 IF Y>3 AND Y<8 THEN 770
690 IF D=203 THEN P=A+1:GOSUB 1290:GOTO 820
700 IF D=237 THEN P=A+1:GOSUB 1290:GOTO 1060
710 IF D=&DD THEN PRINT"INST/IX :":GOTO 1360
720 IF D=&FD THEN PRINT"INST/IY :":GOTO 1360
730 IF D>127 AND D<192 THEN 880
740 IF D<64 THEN 890
750 IF D>191 THEN 1040
760 PRINT:I=1:RETURN
770 IF Z=6 AND Y=7 THEN PRINT"HALT":I=1:RETURN
780 PRINT"LD ";:K=(Y-4)*2
790 IF Z<8 THEN PRINT R$(K);",";
800 IF Z>7 THEN PRINT R$(K+1);",";
810 PRINT R$(Z):I=1:RETURN
820 IF D>63 THEN 840
830 K=INT(D/8):PRINT S$(K);" ";:GOSUB 670:PRINT R$(Z):I=2:RETURN
840 k=INT(d/64)+7:PRINT s$(k);
850 k=INT(d/8)-8:IF d>191 THEN k=k-16
860 IF d>127 AND d<192 THEN k=k-8
870 PRINT k;",";:C$=HEX$(D,2):GOSUB 670:PRINT r$(z):i=2:RETURN
880 k=d-128:w=INT(k/8):PRINT t$(w);:GOSUB 670:PRINT r$(z):i=1:RETURN
890 p=INT(d/10)
900 GOSUB 1170:FOR g=0 TO (d-p*10):READ e$:NEXT g
910 f$=MID$(e$,1,1):IF f$>"9" THEN PRINT e$:i=1:RETURN
920 PRINT MID$(e$,2);
930 IF f$="1" THEN P=A+1:GOSUB 1290:PRINT HEX$(d,2):i=2:RETURN
940 IF F$="2"THEN P=A+2:GOSUB 1290:PRINT HEX$(D,2);:P=A+1:GOSUB 1290:PRIN
T HEX$(D,2):I=3:RETURN
```

```
950 IF f$<>"3" THEN 990
960 P=A+1:GOSUB 1290:PRINT HEX$(d,2);" (";
970 IF d>127 THEN d=d-256
980 w=a+2+d:b$=HEX$(w,4):PRINT b$;")":i=2:RETURN
990 IF f$="4" THEN LOCATE 13,n:f$="2":GOTO 940
1000 IF f$="5" THEN LOCATE 15,n:f$="2":GOTO 940
1010 IF f$="6" THEN LOCATE 13,n:f$="1":GOTO 930
1020 IF f$="7" THEN LOCATE 16,n:f$="2":GOTO 940
1030 PRINT:i=1:RETURN
1040 p=INT((d-192)/20)+7
1050 GOSUB 1170:FOR g=0 TO (d-192-(p-7)*20):READ e$:NEXT g:GOTO 910
1060 IF d<159 THEN 1090
1070 RESTORE 200
1080 FOR j=0 TO d-160:READ e$:NEXT:PRINT e$:i=2:RETURN
1090 IF d>85 THEN 1120
1100 RESTORE 210
1110 FOR j=0 TO d-64:READ e$:NEXT:T=A:A=A+1:GOSUB 910:A=T:i=i+1:RETURN
1120 IF d>114 THEN 1150
1130 RESTORE 650
1140 d=d-22:GOTO 1110
1150 RESTORE 660
1160 d=d-51:GOTO 1110
1170 ON p+1 GOTO 1180,1190,1200,1210,1220,1230,1240,1250,1260,1270,1280
1180 RESTORE 90:RETURN
1190 RESTORE 100:RETURN
1200 RESTORE 110:RETURN
1210 RESTORE 120:RETURN
1220 RESTORE 130:RETURN
1230 RESTORE 140:RETURN
1240 RESTORE 150:RETURN
1250 RESTORE 160:RETURN
1260 RESTORE 170:RETURN
1270 RESTORE 180:RETURN
1280 RESTORE 190:RETURN
1290 IF R=1 THEN D=PEEK(P):RETURN
1300 PH=INT(P/256):PL=P-PH*256
1310 POKE &AB08,PL:POKE &AB09,PH
1320 CALL &AB00
1330 D=PEEK(&AB0E):RETURN
1340 IF Q=0 THEN Q=1:LOCATE#1,24,5:PRINT#1,CHR$(24);"ON";CHR$(24);"/OFF":
RETURN
```

```
1350 Q=0:LOCATE#1,24,5:PRINT#1,"ON/";CHR$(24);"OFF";CHR$(24):RETURN
1360 P=A+1:GOSUB 1290
1370 IF D=9 OR D=&19 OR D=&29 OR D=&39 OR D=&2B OR D=&E3 OR D=&23 OR D=&E
9 OR D=&F9 OR D=&E1 OR D=&E5 THEN I=2:RETURN
1380 IF D=&CB OR D=&36 OR D=&22 OR D=&2A OR D=&21 THEN I=4:RETURN
1390 I=3:RETURN
```

Exemple de desassemblage sur imprimante

```
0FC4    PUSH AF       F5
0FC5    PUSH HL       E5
0FC6    LD A,D        7A
0FC7    CPL           2F
0FC8    LD H,A        67
0FC9    LD A,E        7B
0FCA    CPL           2F
0FCB    LD L,A        6F
0FCC    INC HL        23
0FCD    ADD HL,BC     09
0FCE    INC HL        23
0FCF    EX (SP),HL    E3
0FD0    XOR A         AF
0FD1    SUB E         93
0FD2    PUSH AF       F5
0FD3    CALL 0BA9     CD A9 0B
0FD6    PUSH HL       E5
0FD7    LD A,B        78
```

1

AMSLOG

Stéphane CLOIREC

C'est au MIT que Seymour PAPERT a développé un langage évolué, simple d'emploi, offrant pourtant des possibilités importantes : le LOGO.

Ce langage est surtout orienté vers le graphisme, et est très facile d'accès car chaque instruction LOGO se concrétise dans le déplacement d'une "tortue" à l'écran. L'auteur a alors devant lui une image réelle de ce qu'il vient de créer, et élabore peu à peu son graphisme en s'identifiant à la petite tortue qui avance, tourne, recule sous ses ordres.
Le LOGO est en général très gourmand en mémoire ; c'est pourquoi il est encore assez peu répandu et utilisé. Il est pourtant particulièrement recommandé pour l'apprentissage de l'informatique et de la programmation de par sa simplicité.
Ce programme est une version simplifiée de ce langage ; il permet uniquement de faire du graphisme. Il est programmé exclusivement en Basic : la rapidité d'exécution s'en ressent évidemment, mais cela permettra à tous ceux qui veulent l'améliorer, le modifier, de le faire facilement. AMSLOG comporte deux modes :
— le premier est un mode direct : vous tapez votre instruction dans une fenêtre en bas d'écran, et l'interpréteur Basic l'exécute immédiatement dans la fenêtre graphique. On rentre ainsi ses commandes une à une, et le résultat s'affiche au fur et à mesure ;
— le second est le mode "programme" : vous rentrez les instructions que comporte votre programme les unes après les autres ; vous pouvez ensuite le lister, le modifier ou l'exécuter. Les numéros de ligne s'affichant automatiquement, il n'est pas nécessaire de les taper. Par contre, il est indispensable de bien respecter la syntaxe des instructions, notamment pour les espaces et les paramètres.
Quelques spécifications :
— capacité de programmation : 100 lignes, une instruction par ligne,
— 20 labels utilisés pour les branchements et les sauts (label X),

— 10 variables indicées V(x) [de V(0) à V(9)].
Voici les commandes de l'éditeur :
E : exécute le programme.
L : liste les 20 premières lignes du programme.
LX : liste à partir de la ligne X.
I : insère une ligne en décalant la première ligne du listing à l'écran vers le bas.
D : délète la première ligne du listing à l'écran.
G : permet de passer du mode programme au mode direct, et inversement.
H : donne la liste des instructions et leur syntaxe (Help).
Q : quitte le programme.
SAUVE : sauvegarde le programme LOGO sur cassette.
CHARGE : charge un programme LOGO en mémoire.
VAR V(x) : affiche la valeur de la variable V(x).
Les instructions utilisables sont les suivantes :
AVANCE x : dessine à l'écran un segment de droite de x pixels.
DIRECTION x : la direction courante devient x, x étant un angle en degrés.
DROITE x : tourne à droite de x degrés (de même GAUCHE x tourne à gauche).
ORIGINE xxx.xxx : positionne l'origine graphique en xxx.xxx (attention, il est indispensable de taper les coordonnées en 3 chiffres, séparées par un point. Exemple : ORIGINE 320.200 place le curseur graphique en milieu d'écran).
RECTANGLE xxx.xxx : dessine un rectangle dont les côtés mesurent xxx et xxx pixels (même remarque que pour ORIGINE).
CERCLE x : trace un cercle de rayon x, et centre le dernier point tracé.
EFFACE : efface l'écran.
OUBLIER : efface le programme LOGO contenu en mémoire.
SON : produit un BEEP bref.
MODE x, ENCRE xx.xx, PAPIER x, STYLO x, BORD x : correspondent aux instructions AMSTRAD mode, ink, paper, pen, border (attention au mode employé pour le nombre de couleurs utilisables).
AFFICHE EN xx.xx chaîne de caractères : affiche la chaîne de caractères à la position xx.xx.
V(x) = n : donne à la variable V(x) la valeur n.
CALCUL DE V(x) = V(x) + n et CALCUL de V(x) = V(x) – n : incrémentent ou décrémentent V(x) de la valeur n.
LABEL x : positionne un label sur lequel vont s'effectuer les branchements et les sauts.
VA AU LABEL x : saut inconditionnel au label X.
SI V(x) = 0 VA AU LABEL n : teste si V(x) = 0, et si c'est exact, se rend au Label n.
FIN : obligatoire en fin de programme.

REMARQUES

— Toutes les instructions graphiques sont paramétrables à l'aide des variables V(x),
— à l'origine, AMSLOG, se trouve en mode programme,
— après un L ou un Lx, la ligne courante est la première du listing,
— pour modifier une ligne x, il suffit donc de taper "Lx" puis la nouvelle ligne à la place,
— si jamais le programme se "plante", on peut le redémarrer par un "GOTO 8000" sans perdre le programme LOGO en mémoire,
— de même, il est possible à tout moment de passer du mode direct au mode programme et vice-versa, sans perte du programme LOGO en mémoire.

Enfin, voici quelques exemples de programmes courts et simples, illustrant les possibilités de l'AMSLOG, et qui vous serviront d'exemples pour l'apprentissage du langage.

```
 1 efface
 2 V(1)=36
 3 Label 1
 4 avance 90
 5 droite 110
 6 calcul de V(1)=V(1)-1
 7 si V(1)=0 va au Label 2
 8 va au Label 1
 9 Label 2
10 son
11 fin
```

Exécutez-le, puis entraînez-vous à le modifier (à titre d'exemple, tapez "DROITE 50" au lieu de "DROITE 110").

```
 1 efface
 2 V(1)=25
 3 Label 1
 4 rectangle 100.050
 5 droite 115
```

```
 1 efface
 2 V(1)=36
 3 Label 1
 4 rectangle 100.050
 5 droite 110
```

(La suite de ces deux programmes reste la même que celle du premier.)

```
1 ' *******************************
2 ' ***                       ***
3 ' ***         AMSLOG        ***
4 ' ***                       ***
5 ' ***      PRESENTATION     ***
6 ' ***                       ***
7 ' *******************************
8 '
9 '
10 BORDER 0:INK 0,0
15 CLEAR:MODE 1:RAD
20 DIM PAL(14):FOR I=0 TO 14:READ PAL(I):INK I+1,PAL(I):NEXT
30 DATA 3,6,15,24,25,22,21,12,13,1,2,5,4,14,26
40 MODE 0:CLG 0:PAPER 0:PEN 14
50 ORIGIN 320,250
60 FOR J=48 TO 108:I=J*PI/52:MOVE COS(I)*220,SIN(I)*160-40:DRAW 0,0,J MOD
 13+1:NEXT
70 FOR J=3 TO 23:FOR K=-110 TO 110 STEP 220:I=J*PI/26:MOVE COS(I)*100+K,S
IN(I)*80-20:DRAW 0,0,(J+((K=110)AND 4)-2) MOD 13+1:NEXT K,J
80 LOCATE 8,2:PRINT"AMSLOG":ORIGIN 220,392:DRAWR 197,0:DRAWR 0,-32:DRAWR
-197,0:DRAWR 0,32
90 FOR K=1 TO 10:FOR J=1 TO 13:FOR I=1 TO 13
100 INK I,PAL((J+I)MOD 13):NEXT I,J,K
110 LOCATE 1,25:PRINT"Chargement en cours":RUN"amslog1"
```

```
10 ' ****************************************
20 ' *********                    **********
30 ' *********        AMSLOG      **********
40 ' *********                    **********
50 ' ***                                  ***
60 ' *** (c) 1985  par  Stephane  CLOIREC ***
70 ' ***                                  ***
80 ' ****************************************
90 '
100 '
110 DEG
120 SYMBOL AFTER 165:SYMBOL 166,&6,&8,&3C,&66,&7E,&60,&3C:KEY 131,CHR$(16
6)
```

```
130 GOSUB 2650
140 GOSUB 2280
150 coul=1
160 DIM i$(100)
170 DIM l(20)
180 DIM v(10)
190 GOTO 930
200 '
210 ' *********INTERPRETEUR*********
220 '
230 co=1:d=0
240 IF co>100 THEN 1790
250 c$=i$(co)
260 IF LEFT$ (c$,4)="mode" THEN PEN 1:PAPER 0:MODE VAL(MID$(c$,6)):d=0:za
=320:zb=200:ORIGIN za,zb:GOTO 490
270 IF LEFT$ (c$,4)="bord" THEN GOSUB 2010:GOTO 490
280 IF LEFT$ (c$,5)="encre" THEN GOSUB 2070:GOTO 490
290 IF LEFT$ (c$,6)="papier" THEN GOSUB 2110:GOTO 490
300 IF LEFT$ (c$,5)="stylo" THEN GOSUB 2150:GOTO 490
310 IF LEFT$ (c$,6)="avance" THEN GOSUB 510:GOTO 490
320 IF LEFT$ (c$,6)="droite" OR LEFT$ (c$,6)="gauche" THEN GOSUB 560:GOTO
 490
330 IF LEFT$ (c$,3)="fin" THEN 1790
340 IF LEFT$ (c$,6)="efface" THEN CLS#0:GOTO 490
350 IF LEFT$ (c$,9)="direction" THEN GOSUB 1870:GOTO 490
360 IF LEFT$ (c$,7)="origine" THEN GOSUB 1930:GOTO 490
370 IF LEFT$ (c$,11)="va au label" THEN co=l(VAL (MID$ (c$,13))):GOTO 490
380 IF LEFT$ (c$,2)="v(" THEN v(VAL (MID$ (c$,3,1)))=VAL (MID$ (c$,6)):GO
TO 490
390 IF LEFT$ (c$,5)="si v(" THEN GOSUB 880:GOTO 490
400 IF LEFT$ (c$,9)="calcul de" THEN v(VAL (MID$ (c$,13,1)))=v(VAL (MID$
(c$,13,1)))+VAL (MID$ (C$,20)):GOTO 490
410 IF LEFT$ (c$,5)="liste" THEN p=VAL (MID$ (c$,7)):GOSUB 1270:GOTO 490
420 IF LEFT$ (c$,7)="affiche" THEN LOCATE#gflag,VAL (MID$ (c$,12,2)),VAL
(MID$ (c$,15,2)):PRINT#gflag,MID$(c$,18):GOTO 490
430 IF LEFT$ (c$,3)="son" THEN SOUND 1,89:GOTO 490
440 IF LEFT$ (c$,9)="rectangle" THEN GOSUB 680:GOTO 490
450 IF LEFT$(c$,6)="cercle" THEN GOSUB 1680:GOTO 490
460 IF LEFT$(c$,5)="label" THEN GOTO 490
470 IF gflag=1 THEN 2190
480 GOTO 1610
490 co=co+1
```

```
500 GOTO 240
510 '
520 ' *********AVANCE*********
530 '
540 IF MID$ (c$,8,1)="v" THEN xa=SIN (d)*v(VAL (MID$ (c$,10,1))):xb=COS (
d)*v(VAL (MID$ (c$,10,1))):DRAW xa,xb,coul:za=xa+za:zb=xb+zb:ORIGIN za,zb
:RETURN
550 xa=SIN (d)*VAL (MID$ (c$,8)):xb=COS (d)*VAL (MID$ (c$,8)):DRAW xa,xb,
coul:za=xa+za:zb=xb+zb:ORIGIN za,zb:RETURN
560 '
570 ' *********DROITE/GAUCHE*********
580 '
590 IF MID$(c$,8,1)="v" THEN GOTO 640
600 IF LEFT$ (c$,1)="d" THEN d=d+(VAL (MID$ (c$,8))):RETURN
610 IF LEFT$ (c$,1)="g" THEN d=d-(VAL (MID$ (c$,8))):RETURN
620 GOSUB 1650
630 RETURN
640 IF LEFT$ (c$,1)="d" THEN d=d+v(VAL (MID$ (c$,10,1))):RETURN
650 IF LEFT$ (c$,1)="g" THEN d=d-v(VAL (MID$ (c$,10,1))):RETURN
660 GOSUB 1650
670 RETURN
680 '
690 ' *********RECTANGLE*********
700 '
710 IF MID$(c$,11,1)="v" OR MID$(c$,15,1)="v" THEN GOTO 770
720 l0=VAL (MID$ (c$,11,3)):l1=VAL (MID$ (c$,15))
730 ORIGIN za,zb:xa=SIN(d)*l0:xb=COS(d)*l0:DRAW xa,xb,coul:za=za+xa:zb=zb
+xb:ORIGIN za,zb
740 d=d+90:xa=SIN(d)*l1:xb=COS(d)*l1:DRAW xa,xb,coul:za=za+xa:zb=zb+xb:OR
IGIN za,zb
750 d=d+90:xa=SIN(d)*l0:xb=COS(d)*l0:DRAW xa,xb,coul:za=za+xa:zb=zb+xb:OR
IGIN za,zb
760 d=d+90:xa=SIN(d)*l1:xb=COS(d)*l1:DRAW xa,xb,coul:za=za+xa:zb=zb+xb:OR
IGIN za,zb:d=d+90:RETURN
770 IF MID$(c$,11,1)="v" THEN IF MID$(c$,16,1)="v" THEN l0=v(VAL(MID$(c$,
13,1))):l1=v(VAL(MID$(c$,18,1))) ELSE l0=v(VAL(MID$(c$,13,1))):l1=VAL (MI
D$ (c$,16))
780 IF MID$(c$,15,1)="v" THEN l0=VAL (MID$ (c$,11,3)):l1=v(VAL(MID$(c$,17
,1)))
790 GOTO 730
800 '
810 ' *********LABELS*********
```

```
820
830 PRINT "PREPARATION DES LABELS"
840 FOR n=1 TO 100
850 IF LEFT$ (i$(n),5)="label" THEN l(VAL (MID$ (i$(n),7)))=n
860 NEXT n
870 GOTO 200
880 '
890 ' *********TEST*********
900 '
910 IF v(VAL (MID$ (c$,6,1)))=0 THEN co=l(VAL (MID$ (c$,23)))
920 RETURN
930 '
940 ' *********EDITEUR*********
950 '
960 GOSUB 2280
970 p=1
980 INPUT rep$:c$=LOWER$(rep$):IF c$="" THEN GOTO 980
990 t=0:c$=c$+" ":za=320:zb=200:ORIGIN za,zb:d=0
1000 IF c$="sauve " THEN GOTO 2370
1010 IF c$="charge " THEN GOTO 2510
1020 GOTO 1080
1030 i$(p)=c$
1040 IF c$="oublier " THEN RUN
1050 PRINT CHR$(11)CHR$(11)p;" ";i$(p):PRINT SPC(79):PRINT
1060 p=p+1:IF p=101 THEN p=100
1070 GOTO 980
1080 '
1090 ' *********ORDRES*********
1100 '
1110 IF LEFT$ (c$,2)="l "THEN GOSUB 1210
1120 IF c$="i " THEN GOSUB 1350
1130 IF c$="d " THEN GOSUB 1450
1140 IF c$="e " THEN GOTO 800
1150 IF c$="q " THEN END
1160 IF c$="g " THEN GOTO 2190
1170 IF C$="h "THEN GOSUB 2960:GOTO 930
1180 IF LEFT$ (c$,3)="var" THEN GOSUB 1530
1190 IF t=1 THEN GOTO 980
1200 GOTO 1030
1210 '
1220 ' *********LISTE*********
1230 '
```

```
1240 CLS:IF LEN (c$)=2 THEN c$=c$+"1"
1250 p=VAL (MID$ (c$,3))
1260 IF p>100 THEN GOTO 1320
1270 FOR n=p TO p+19
1280 IF n>100 THEN GOTO 1320
1290 IF LEFT$ (i$(n),3)="fin" THEN PRINT:PRINT" ";i$(n);"du programme.":G
OTO 1320
1300 PRINT n;:IF n<10 THEN PRINT" ";i$(n) ELSE PRINT i$(n)
1310 NEXT n
1320 t=1
1330 LOCATE 1,24
1340 RETURN
1350 '
1360 ' *********INSERTION*********
1370 '
1380 t=1
1390 FOR n=99 TO p STEP -1
1400 i$(n+1)=i$(n)
1410 NEXT n
1420 i$(p)=">"
1430 c$="1 "+STR$ (p):GOSUB 1240
1440 RETURN
1450 '
1460 ' *********DELETE*********
1470 '
1480 FOR n=p TO 99
1490 i$(n)=i$(n+1):NEXT n
1500 i$(100)=" "
1510 c$="1 "+STR$ (p):GOSUB 1240
1520 RETURN
1530 '
1540 ' *********VARIABLES*********
1550 '
1560 CLS
1570 PRINT "v(";MID$(c$,7,1);")=";v(VAL (MID$ (c$,7,1))):t=1:RETURN
1580 '
1590 ' *********ERREURS********
1600 '
1610 CLS:PRINT "Instruction: ";c$:PRINT "Erreur en ligne ";co:FOR pa=1 TO
 2000:NEXT pa:c$="1 "+STR$ (co):GOTO 990
1620 '
1630 ' ********ANGLES********
```

```
1640 '
1650 IF d>360 THEN d=d-360
1660 IF d<0 THEN d=d+360
1670 RETURN
1680 '
1690 ' *********CERCLE*********
1700 '
1710 IF MID$(c$,8,1)="v" THEN 1770
1720 r=VAL(MID$(c$,8))
1730 FOR i=0 TO 360
1740 PLOT r*SIN(i),r*COS(i),coul
1750 NEXT i
1760 RETURN
1770 r=v(VAL(MID$(c$,10,1)))
1780 GOTO 1730
1790 '
1800 ' *********COMMANDE*********
1810 '
1820 INK 1,0:PEN 1:LOCATE 1,25:INPUT"COMMANDE ---> ",rep$:c$=LOWER$(rep$)
1830 IF c$="e" THEN GOSUB 2280:GOTO 800
1840 IF LEFT$(c$,1)="l"THEN c$=c$+" ":GOSUB 2280:GOTO 1080
1850 IF c$="q" THEN GOSUB 2280:END
1860 LOCATE 1,25:PRINT SPC(78):GOTO 1790
1870 '
1880 ' *********DIRECTION/ORIGINE*********
1890 '
1900 IF MID$(C$,11,1)="v" THEN 1920
1910 d=VAL (MID$ (c$,11)):RETURN
1920 d=v(VAL(MID$(c$,13,1))):RETURN
1930 '
1940 IF MID$(c$,9,1)="v" OR MID$(c$,13,1)="v" THEN 1980
1950 za=VAL (MID$ (c$,9,3))
1960 zb=VAL (MID$ (c$,13,3))
1970 ORIGIN za,zb:RETURN
1980 IF MID$(c$,9,1)="v" THEN IF MID$(c$,14,1)="v" THEN za=v(VAL (MID$ (c
$,11,1))):zb=v(VAL (MID$ (c$,16,1))) ELSE za=v(VAL (MID$ (c$,11,1))):zb=V
AL (MID$ (c$,14,3))
1990 IF MID$(c$,13,1)="v" THEN za=VAL (MID$ (c$,9,3)):zb=v(VAL (MID$ (c$,
15,1)))
2000 ORIGIN za,zb:RETURN
2010 '
2020 ' *********BORD/ENCRE/PAPIER/STYLO*********
```

```
2030
2040 IF MID$(c$,6,1)="v" THEN 2060
2050 BORDER VAL(MID$(c$,6)):RETURN
2060 BORDER v(VAL(MID$(c$,8,1))):RETURN
2070 '
2080 IF MID$(c$,10,1)="v" THEN GOTO 2100
2090 INK VAL(MID$(c$,7,2)),VAL(MID$(c$,10)):RETURN
2100 INK VAL(MID$(c$,7,2)),v(VAL(MID$(c$,12,1))):RETURN
2110 '
2120 IF MID$(c$,8,1)="v" THEN GOTO 2140
2130 PAPER#gflag,VAL(MID$(c$,8)):CLS:RETURN
2140 PAPER#gflag,v(VAL(MID$(c$,10,1))):CLS:RETURN
2150 '
2160 IF MID$(c$,7,1)="v" THEN GOTO 2180
2170 PEN#gflag,VAL(MID$(c$,7)):coul=VAL(MID$(c$,7)):RETURN
2180 PEN#gflag,v(VAL(MID$(c$,9,1))):coul=v(VAL(MID$(c$,9,1))):RETURN
2190 '
2200 ' **********MODE DIRECT*********
2210 '
2220 GFLAG=1
2230 CLS:WINDOW 1,80,23,25:PAPER 1:PEN 0:CLS
2240 PRINT:INPUT "Commande --> ",rep$:c$=LOWER$(rep$)
2250 IF c$="efface" THEN PEN 1:PAPER 0:MODE 2:GOTO 2190
2260 IF c$="g" THEN GOSUB 2280:gflag=0:GOTO 930
2270 GOTO 260
2280 '
2290 ' ********INITIALISATION********
2300 '
2310 MODE 2
2320 za=320:zb=200:d=0
2330 ORIGIN za,zb
2340 INK 1,0:INK 0,13
2350 PEN 1:PAPER 0:BORDER 13
2360 RETURN
2370 '
2380 ' ********SAUVEGARDE********
2390 '
2400 CLS:PRINT"SAUVEGARDE DU PROGRAMME"
2410     PRINT"-----------------------"
2420 LOCATE 1,10:INPUT"Entrez le nom de votre programme : ",nom$
2430 LOCATE 1,18:PRINT" Veuillez patienter,"
2440             PRINT"Sauvegarde en cours...":PRINT:PRINT
```

```
2450 OPENOUT nom$
2460 FOR bc=1 TO 100
2470 PRINT#9,i$(bc)
2480 IF i$(bc)="fin"THEN bc=101
2490 NEXT bc
2500 CLOSEOUT:GOTO 930
2510 '
2520 ' ********CHARGEMENT********
2530 '
2540 CLS:PRINT"CHARGEMENT DU PROGRAMME"
2550      PRINT"-----------------------"
2560 LOCATE 1,10:INPUT"Entrez le nom de votre programme : ",nom$
2570 LOCATE 1,18:PRINT" Veuillez patienter."
2580 PRINT"Chargement du programme ";nom$;" en cours...":PRINT:PRINT
2590 OPENIN "
2600 FOR bc=1 TO 100
2610 INPUT#9,i$(bc)
2620 IF i$(bc)="fin" THEN bc=101
2630 NEXT bc
2640 CLOSEIN:GOTO 930
2650 '
2660 ' *********PRESENTATION*********
2670 '
2680 MODE 0:INK 3,1,5:INK 1,1:INK 0,0:BORDER 0
2690 ch$="AMSLOG"
2700 ORIGIN 186,347:DRAWR 204,0,1
2710 DRAWR 0,-34:DRAWR -202,0:DRAWR 0,34
2720 PEN 3
2730 FOR i=1 TO 6
2740 cd$=MID$(ch$,i,1)
2750 LOCATE 6+i,5:PRINT cd$
2760 SOUND 1,89+i*10
2770 FOR pa=1 TO 100:NEXT pa,i
2780 PEN 1
2790 LOCATE 2,15:PRINT"Langage Graphique"
2800 LOCATE 1,17:PRINT"Etudi@ pour le D.A.O"
2810 INK 2,3:PEN 2:LOCATE 2,20:PRINT CHR$(164)+" St@phane CLOIREC"
2820 LOCATE 2,24:INK 4,9:PEN 4:PRINT"Instructions (O/N)?"
2830 rep$=INKEY$:r$=LOWER$(rep$):IF r$="n" THEN RETURN:IF r$="o" THEN GOT
O 2860
2840 IF r$="o" THEN GOTO 2860
2850 GOTO 2830
```

```
2860 MODE 1
2870 INK 1,11:PEN 1
2880 LOCATE 5,1:PRINT "***** AMSLOG:Instructions *****"
2890 LOCATE 1,4:PRINT"   Ce programme est une adaptation sim- -plifi@e du
 langage LOGO."
2900 PRINT:PRINT:PRINT"   Il comporte deux modes diff@rents:          -
 un mode direct (l'ordinateur interpr@te imm@diatement les ordres que vou
s lui fournissez)"
2910 PRINT:PRINT"          - un mode programme (il stocke alors les instru
ctions sous forme d' un listing pour ex@cution ult@rieure)"
2920 PRINT:PRINT:PRINT"   Un @diteur de texte vous permet ausside modifie
r vos programmes (supprimer etins@rer des lignes, lister, ex@cuter...)"
2930 LOCATE 1,24:PRINT "Tapez une touche":LOCATE 35,24:INK 2,11,0:PEN 2:P
RINT "-->"
2940 r$=INKEY$:IF r$<>"" THEN GOTO 2960
2950 GOTO 2940
2960 MODE 1:PEN 1
2970 PRINT "   Voici la liste des instructions:"
2980 PRINT:PRINT:PRINT"H :affiche les instructions utilisables":PRINT"L :
liste les 22 premieres lignes.        L x:liste a partir de la ligne x."
2990 PRINT"I :insere un espace entre deux lignes en   d@calant la 1ere li
gne du dernier        listing vers le bas."
3000 PRINT"D :supprime la 1ere ligne du dernier       listing et remonte
toutes les autres    d'une ligne.
3010 PRINT"E :ex@cute un programme."
3020 PRINT"Q :quitte le programme."
3030 PRINT"SAUVE :sauvegarde le programme sur k7."
3040 PRINT"CHARGE :lit un programme sur k7."
3050 PRINT"VAR V(x) :affiche la valeur de la                variable v(x
)"
3060 PRINT"OUBLIER :perd le programme"
3070 LOCATE 1,24:PRINT"Tapez une touche":LOCATE 35,24:PEN 2:PRINT"-->"
3080 r$=INKEY$:IF r$<>""THEN 3100
3090 GOTO 3080
3100 CLS:PEN 1
3110 PRINT" Mode programme."
3120 PRINT:PRINT"V(x)=n :donne a la variable v(x) la           valeur n
."
3130 PRINT"DIRECTION n :la direction en cours                  devient n
."
3140 PRINT"ORIGINE nnn,nnn:l'origine du graphisme                 est pl
ac@e au coordon-                 -n@es nnn,nnn"
```

```
3150 PRINT"EFFACE :efface l'@cran."
3160 PRINT"ENCRE xx.nn :donne a l'encre xx la va-                -leur nn.
"
3170 PRINT"PAPIER xx :le papier prend la couleur de           l'encre xx.
"
3180 PRINT"STYLO xx :le stylo prend la couleur de              l'encre xx."
3190 PRINT"BORD nn :le bord de l'@cran prend la               couleur nn."
3200 LOCATE 1,24:PRINT"Tapez une touche":LOCATE 35,24:PEN 2:PRINT"-->"
3210 r$=INKEY$:IF r$<>""THEN 3230
3220 GOTO 3210
3230 CLS:PEN 1
3240 PRINT"AFFICHE EN nn,nn :@crit en nn.nn ce qui                        est
affich@ a la ligne                  suivante du programme."
3250 PRINT"LABEL n :positionne un label (utilis@            pour les saut
s)"
3260 PRINT"SON :fait un son bref et aigu"
3270 PRINT"AVANCE x :avance de x pixels dans la               direction en
 cours."
3280 PRINT"GAUCHE x :tourne a gauche de x degr@s."
3290 PRINT"DROITE x :tourne a droite de x degr@s."
3300 PRINT"FIN :fin du programme."
3310 PRINT"VA AU LABEL x :saute au label x."
3320 PRINT"SI V(x)=0 VA AU LABEL x :saut au label x
   si v(x) @gale 0"
3330 PRINT"LISTE x :liste le programme a partir de          la ligne x."
3340 PRINT"CALCUL DE V(x)=... :v(x) prend la valeur                      de
 ce qui suit le                        signe @gal."
3350 LOCATE 1,24:PRINT"Tapez une touche":LOCATE 35,24:PEN 2:PRINT"-->"
3360 r$=INKEY$:IF r$<>""THEN 3380
3370 GOTO 3360
3380 CLS:PEN 1
3390 PRINT"RECTANGLE xxx.xxx :trace un rectangle de                       cot
@s xxx et xxx."
3400 PRINT"CERCLE xxx :trace un cercle de rayon xxx"
3410 PRINT:PRINT:PRINT"   Tous les trac@s ont pour origine le  dernier po
int inscrit.                    De plus,il est possible de param@trertou
tes les instructions avec les varia- -bles v(x)."
3420 PRINT:PRINT:PRINT"   Vous aller maintenant pouvoir commen--cer a pro
grammer en AMSLOG.Vous verrez qu'il est possible,avec de tres courts  pro
grammes,de cr@er de superbes graphis--mes."
3430 LOCATE 1,25:PRINT "Tapez une touche":LOCATE 35,25:PEN 2:PRINT "-->"
3440 IF INKEY$<>"" THEN RETURN
3450 GOTO 3440
```

2

LA SAUVEGARDE DES DONNÉES

Michel ARCHAMBAULT

Une cassette vierge dans le magnéto-cassette ne sert pas uniquement à sauver un programme que vous avez écrit, on peut aussi enregistrer des valeurs (nombres, chaînes) que l'on tient à conserver : c'est alors un enregistrement de fichier.

C'est un peu moins simple que le SAVE " " pour un programme Basic, mais ce n'est pas compliqué à partir de l'instant où l'on a compris la raison d'être de certaines fonctions du genre OPENIN, CLOSEOUT, etc.

LES DIRECTIONS IN ET OUT

Le micro-ordinateur pense par rapport à lui-même ; le magnéto, ce n'est pas *dans sa mémoire,* il fait partie de l'extérieur (même si le constructeur l'a mis sous le même capot en plastique). Le clavier, l'écran, une imprimante, un lecteur de disquettes sont aussi, pour lui, des éléments extérieurs (on les appelle aussi des "périphériques").
Quand il envoie sur cassette, c'est OUT ; quand il reçoit des données de la cassette, c'est du IN. D'accord ?
Ce n'est pas la peine de le spécifier pour le clavier (toujours du IN), ou pour l'écran et une imprimante (toujours du OUT), uniquement pour les périphériques bi-directionnels, à savoir magnéto-cassette, lecteur-enregistreur de disquettes (ou "DRIVE") et MODEM.
La nature de l'ordre Basic le prévient sur le sens : PRINT ou WRITE, c'est du OUT ; INPUT, c'est du IN. Encore faut-il lui dire le nom du périphérique expéditeur ou récepteur, c'est le "canal", #0 à #9 (le signe # se prononce dièse). Quand on ne précise pas le canal, c'est le canal #0 ; ainsi PRINT #0 = vers l'écran, INPUT #0 = entrée venant du clavier, le canal zéro est donc le canal implicite.
Le canal #8, c'est l'imprimante, on écrit donc simplement PRINT #8.

Le canal cassette, c'est # 9. Alors là, tout se complique, car il faut *prévenir* l'ordinateur s'il lui faut OUVRIR sa mémoire pour une sortie ou pour une entrée : c'est OPENOUT ou OPENIN. A la fin du transfert de données, il faut le prévenir que c'est terminé, c'est CLOSEOUT ou CLOSEIN.

UN EXEMPLE SIMPLE

Un programme vous a fait entrer (INPUT) des noms, des nombres ; il a calculé des résultats. Vous voulez conserver tout cela sur bande magnétique. Ce fichier s'appellera "MACHIN" (original, n'est-ce pas ?).

```
400 OPENOUT "MACHIN"
410 PRINT #9, A$, B$, C$, NB, TOTAL, FZ$
430 CLOSEOUT
```

Le nom du fichier n'est pas obligatoire, onn peut aussi commander OPENOUT " ".
A la suite de PRINT # 9, on retrouve la même écriture que PRINT à l'écran ; les virgules sont des "séparateurs" de zones de données, comme à l'écran, comme dans une ligne de DATA, comme une suite de paramètres après une fonction Basic.
Qu'y a-t-il sur la bande enregistrée ? Un signal de début fichier, le nom du fichier, trois chaînes, puis deux nombres, une chaîne, et enfin le signal de fin de fichier. Il n'y a pas les noms des variables enregistrées, seulement leurs valeurs. Ce point est très important.
Complétons un peu. Enregistrons le contenu d'un tableau DIM F(100,5). Insérons la ligne 420 :

```
420 FOR I=1 TO 100:FOR J=0 TO 5:PRINT #9, F(I,J):NEXT:NEXT
```

Une seule ligne pour sauvegarder un tableau de 600 nombres ! Pas mal...
Résumons la technique : après le OPENOUT " ", on fait PRINT # 9 sur chaque valeur à enregistrer. On ferme par CLOSEOUT.

RETROUVONS NOS VALEURS

Pour mettre en mémoire vive (en RAM) ces valeurs sur bande, il suffit d'écrire ces mêmes lignes en Basic, mais en remplaçant OUT par IN et PRINT # 9 par INPUT # 9. C'est tout ; mais attention à trois pièges hélas classiques :
— demander un nombre alors que c'est une chaîne qui se présente ou vice-versa. Ainsi, nous avions, dans l'ordre, A$, B$, C$, NB, etc. Si nous programmons :

INPUT #9, A$, B$, C, NB
il y aura un plantage. Mais, on a droit à :
INPUT #9, NOM$, VILLE$, REP$, PRIX, etc. ;
— on demande d'entrer un nombre de variables plus grand ou plus petit que contient le fichier sur bande : plantage ;
— dans le cas de "variables indicées" (celles d'un tableau DIM), on a oublié d'annoncer ce DIM. Exemple : DIM F(100,5) en *début* du programme. Là aussi plantage.

LE CAS DES FICHIERS ELASTIQUES

C'est-à-dire des fichiers dont le nombre de valeurs peut varier d'un enregistrement à un autre. Le cas est fréquent, mais il y a deux façons de se tirer d'affaire : par la fonction EOF (=END OF FILE) qui signale qu'il a reçu le signal de fin de fichier, ou encore en prévenant en début de fichier du nombre de valeurs qui vont suivre. Voyons un exemple avec EOF.

```
500 OPENIN "TRUC"
510 IF EOF THEN 550
520 INPUT #9, AD, NB
530 POKE AD, NB
540 GOTO 510
550 CLOSEIN
```

Notez bien que la boucle de chargement 510-540 débute par le texte EOF, ce qui évite tout problème.
Second exemple : à l'enregistrement, nous avions partiellement garni un tableau de noms prévu pour un maximum de 500, soit DIM NOM$(500). En fait, on n'en a entré aujourd'hui que 48. Le programme les a comptés, c'est la variable NB=48. Le fichier a été enregistré comme suit :

```
800 OPENOUT "REUNION DU 28/6/86"
810 PRINT #9, NB
820 FOR I=1 TO NB
830 PRINT #9, NOM$(I):NEXT
840 CLOSEOUT
```

Dans le sous-programme de chargement :

```
910 INPUT "TITRE", NDF$
920 OPENIN NDF$
930 INPUT #9, NB
940 FOR I=1 TO NB
950 INPUT #9, NOM$(I):NEXT
960 CLOSEIN
```

Et le tour est joué. Facile, non ?

CONCLUSION

Ces quelques mots Basic que nous espérons avoir démystifiés sont d'une importance primordiale : ils permettent d'archiver des données entrées ou calculées par l'ordinateur, et ce mieux que sur papier, car elles peuvent redevenir actives après rechargement sans que l'on ait eu à les lire sur une feuille ou à les retaper au clavier. Un micro-ordinateur, c'est aussi autre chose qu'une super calculatrice programmable...
Vous avez ainsi tous les éléments pour vous confectionner des fichiers sur bande à vos mesures.
Lancez-vous. OPEN !

TRACEUR DE CAMEMBERTS

Michel ARCHAMBAULT

Les résultats d'élections à la télévision nous ont familiarisés avec cette représentation circulaire des divers pourcentages obtenus. C'est beaucoup plus "parlant" que les diagrammes en barres verticales.

Le programme que nous vous proposons est complet, à savoir : la saisie des données brutes, le calcul automatique en pourcentages, le classement (facultatif), l'affichage de la représentation graphique en camembert avec la liste complète des éléments (pourcentages et total), édition sur imprimante de la liste des résultats et "Hard copy" de la page graphique. Ajoutez à cela les corrections d'erreurs à la saisie et une multitude de sécurités anti-étourderies. Un logiciel d'une rapidité étonnante pouvant être utilisé à des fins professionnelles, et qui ne fait que 3655 octets REM comprises, soit deux "Blocks" sur cassette.

LE MODE D'EMPLOI

Il est parfois beaucoup plus clair de commencer par la fin...
La première page d'écran est le titre qui vous demande d'entrer le nom de baptême de votre tableau. Tous les caractères tels virgule et guillemets sont admis. Un pointillé vous indique le nombre de caractères maxi (40).
Deuxième page d'écran : la saisie des données brutes. L'en-tête rappelle que vous avez droit à douze noms (ou éléments) de 15 caractères maxi chacun. La numérotation de 1 à 12 est automatique. Nous sommes toujours en MODE 1.
Entrez le premier nom, ENTER, quantité correspondante, ENTER, nom suivant, etc.
En bas de l'écran figurent deux options : E pour erreur, Q pour quitter la saisie (automatique après le douzième nom). Deux cas d'erreurs :

— le curseur étant sur ''quantité'', vous constatez une erreur sur le nom situé à gauche (vous avez déjà pressé ENTER). Tapez E,ENTER; vous pouvez réécrire le nom ;
— le curseur est sur ''NOM'' et vous constatez que la ligne supérieure est erronée, tapez E et vous revenez sur la ligne précédente.
A la fin de la saisie, on vous demande si vous voulez un classement. Répondons N (non), par exemple. Aussitôt après...
Troisième page d'écran, en MODE 2, le camembert se dessine à gauche. Chaque secteur est repéré par des lettres A, B, C, D, etc. En haut apparaît votre titre. A droite la légende, Lettre A = tel nom = tel pourcentage, etc. Au-dessous ''Sur un total de ...'' (somme des quantités). En bas à droite, le choix entre quatre options :

I = I IMPRIMER

(Si vous avez une imprimante...). Il y a d'abord écriture de la liste détaillée des résultats, saut de page paramétré et copie sur imprimante (ici EPSON ou assimilée) de la page d'écran graphique. Un bip sonore signale la fin de ce travail. Les autres options sont toujours disponibles.

C = CLASSER

La page d'écran est modifiée avec, cette fois-ci, le plus fort pourcentage obtenu en ''A'', et ainsi de suite. On peut commander une deuxième impression.

E = ENCORE

C'est un RUN du programme pour un autre tableau.

F = FIN

Retour à l'état initial de l'AMSTRAD (couleurs), MODE 1 et END.

LE PROGRAMME

Si vous n'avez pas d'imprimante, vous pouvez vous dispenser de taper les lignes 4220, 5000 à 5110, et de 56000 à la fin. De même, supprimez Z (200) en ligne 50.
Pour beaucoup d'entre vous, ce sera l'occasion d'utiliser certaines fonctions Basic peu courantes mais tellement pratiques, qui font la gloire de l'AMSTRAD.

— UPPER$ qui transforme tout en majuscules.
— STRING$ qui évite de taper une longue série du même caractère ; de même SPC pour les blancs (dont le manuel ne parle pas...).
— CHR$(18) qui efface la ligne à droite du curseur.
— CHR$(7) qui émet un bip sonore.
— XPOS, YPOS qui permettent de faire un LOCATE à l'extrémité d'un DRAW ou d'un DRAWR.
— LINE INPUT qui est à INPUT ce que WRITE est à PRINT (prend absolument TOUS les caractères).

Nous avons ajouté une fonction Basic manquante : CIRCLE. C'est notre mini-module 54000 à 54040 qui trace un cercle parfait (pas en pointillés...) en une seconde ! On lui donne les coordonnées X, Y de son centre, son rayon R et sa couleur COL (PEN 1 par défaut). Son principe est de tracer 36 "cordes".
C'est arbitrairement que nous avons limité le nombre de noms à 12. Il ne faut pas confondre camembert et roue de bicyclette !
Nous avons souvent usé d'un côté sympa de l'AMSTRAD qui, lorsqu'il attend des valeurs entières (exemple dans un LOCATE) et qu'on lui livre des valeurs calculées décimales, transforme ces dernières en entiers et arrondis au plus proche !

TABLEAUX DE CALCULS

DIM CA(12,3) : En 1 ce sont les quantités NB saisies (totalisées par TNB). En 2, le rapport NB/TNB (multiplié par 100 = %). En 3 = la position angulaire, donc tenant compte des angles des secteurs antérieurs. Ils sont recalculés si on a demandé un classement (tri) en deuxième passe.

TRACE

Après tracé des rayons séparant les secteurs (ligne 4040), on trace "à blanc" les bissectrices de ces secteurs, mais avec un rayon plus court (lignes 4050 à 4080). Il s'agit alors de transformer ces coordonnées graphiques (XPOS et YPOS) en coordonnées texte (LOCATE) afin d'inscrire l'initiale du secteur (A, B, C...). Le calcul de la ligne 4090 fait une division par 16 pour l'axe Y (400/25 = 16) et par 8 pour l'axe des X (640/80 = 8), car nous sommes ici en MODE 2.
La rapidité d'exécution est assez surprenante.

IMPRESSION

Le module 5000-5110 peut être appliqué à toute imprimante, à l'exception du CHR$(27)"   " (ligne 5010) qui est un code EPSON qui vide

le buffer de l'imprimante (= ''initialisation d'imprimante''). Une précaution systématique. Le saut de page de la ligne 5090 s'applique au papier 11 pouces de 66 lignes par page (le plus courant).
Le module de Hard-Copy (56000-56240) ne concerne que les imprimantes EPSON RX ou FX (et les EPSON vendues sous une quinzaine d'autres marques...). Le principe consiste à explorer la mémoire d'écran en 80 bandes verticales, mise en DIM Z de ces octets, et envoi de ceux-ci en ''BIT IMAGE'' (code CHR$(27) ''K'' de la ligne 56120). On obtient alors un tracé 18 × 24 cm qui déforme légèrement le cercle.

NOTE : Si vous utilisez ce module utilitaire pour d'autres programmes, n'oubliez pas le DIM Z (200) mis ici en ligne 50.

```
10 ' CAMEMBERT / AMSTRAD
20 ' Michel ARCHAMBAULT 1985
50 DIM NM$(12),CA(12,3),Z(200)
60 'DIM CA:1=QUANTITE NB;2=NB/TOTAL;3=POSITION ANGULAIRE
70 INK 0,1:INK 1,24:INK 2,20:INK 3,6:BORDER 15:ORIGIN 0,0
1000 'TITRE
1010 MODE 1:PAPER 2:PEN 3:CLS:LOCATE 20,9:PRINT " C A M E M B E R T"
1020 R=120:X=160:Y=260:COL=-1:GOSUB 54000
1030 LOCATE 3,20:PEN 0:PRINT "Nom du Tableau: ";STRING$(22,".");STRING$(1
8,"."):PEN 3:LOCATE 19,20:LINE INPUT TIT$
1040 TIT$=LEFT$(TIT$,40)
1100 'SAISIE
1110 CLS:LOCATE 3,2:PRINT "12 NOMS maxi de 15 CARACTERES maxi.
1120 LOCATE 4,24:PRINT "Erreur = E ; pour Quitter NOM = Q"
1130 LOCATE 10,4:PRINT "NOM:":LOCATE 25,4:PRINT "QUANTITE:":PEN 0
1140 N=N+1:LOCATE 5,5+N:PRINT USING "##";N:LOCATE 7,5+N:PRINT " : ";STRIN
G$(15,"."):LOCATE 10,5+N:LINE INPUT "",NM$
1150 IF UPPER$(NM$)="Q" THEN N=N-1:GOTO 2000
1155 IF NM$="E" THEN N=N-2:GOTO 1140
1160 NM$(N)=LEFT$(NM$,15)
1170 LOCATE 27,5+N:INPUT "",NB$:NB=VAL(NB$):IF UPPER$(NB$)="E" THEN N=N-1
:GOTO 1140
1175 IF (NB=0 AND NB$<>"0") OR NB<0 THEN PRINT CHR$(7):GOTO 1170
1180 CA(N,1)=NB:TNB=TNB+NB:IF N=12 THEN 2000
1190 GOTO 1140
2000 'FIN DE SAISIE
2010 PEN 3:IF N=1 THEN PRINT CHR$(7):LOCATE 3,22:PRINT "Pas de Camembert
pour UNE variable...":FOR I=1 TO 3000:NEXT:RUN
2020 LOCATE 6,22:PRINT "Voulez-vous classer ? ( O/N )"
2025 Q$=UPPER$(INKEY$):IF Q$="N" THEN 3000
2030 IF Q$<>"O" THEN 2025
```

```
2040 LOCATE 6,22:PRINT CHR$(18);SPC(11);"Patience..."
2100 'TRI
2110 FOR J=1 TO N:E=CA(J,1)
2120 FOR I=J TO N
2130 IF CA(I,1)>=E THEN E=CA(I,1):K=I:E$=NM$(I)
2140 NEXT
2150 CA(K,1)=CA(J,1):CA(J,1)=E
2160 NM$(K)=NM$(J):NM$(J)=E$
2170 NEXT
3000 'CALCULS DIM CA
3010 FOR I=1 TO N:CA(I,2)=CA(I,1)/TNB:CA(I,3)=360*CA(I,2)+CA(I-1,3):NEXT
4000 ' TRACE
4010 MODE 2:INK 0,13:INK 1,0:BORDER 9
4020 X=180:Y=200:R=170:COL=0:GOSUB 54000
4030 FOR I=1 TO N:PLOT X,Y,1
4040 DRAWR COS(CA(I,3))*R,SIN(CA(I,3))*R:NEXT
4050 FOR I=N TO 1 STEP-1
4060 AB=(CA(I,3)+CA(I-1,3))/2:'ANGLE BISSECTRICE
4070 PLOT X,Y
4080 DRAWR COS(AB)*120,SIN(AB)*120,0
4090 LOCATE XPOS/8,(400-YPOS)/16:PRINT CHR$(64+I):NEXT:PLOT X,Y,1
4100 LOCATE 40+(40-LEN(TIT$))/2,2:PRINT TIT$
4110 FOR I=1 TO N
4120 LOCATE 47,4+I:PRINT CHR$(64+I);" : ";NM$(I)
4130 LOCATE 67,4+I:PRINT "= ";:PRINT USING "##.##";CA(I,2)*100;:PRINT " %
"

4140 NEXT
4150 LOCATE 50,19:PRINT "SUR UN TOTAL DE";TNB
4200 LOCATE 38,23:PRINT "Encore,Fin,Imprimer,Classer ? ( E,F,I,C )"
4210 Q$=UPPER$(INKEY$):IF Q$="E" THEN RUN
4220 IF Q$="I" THEN GOSUB 5000:PRINT#8,CHR$(27)"@":PRINT CHR$(7):GOTO 420
0
4225 IF Q$="C" THEN 2100
4230 IF Q$<>"F" THEN 4210
4240 MODE 1:INK 0,1:INK 1,24:BORDER 1:PAPER 0:PEN 1:CLS
4250 LOCATE 11,12:PRINT "VOUS POUVEZ ETEINDRE.":PRINT:PRINT
4260 END
5000 'IMPRESSION (EPSON)
5010 PRINT#8,CHR$(27)"@";SPC((40-LEN(TIT$))/2);TIT$:PRINT#8
5020 FOR I=1 TO N
5030 PRINT#8,USING "##";I;:PRINT#8," : ";
5040 PRINT#8,NM$(I);SPC(15-LEN(NM$(I)));" = ";
```

```
5050 PRINT#8,USING"##.##";CA(I,2)*100;
5060 PRINT#8," % (";CA(I,1);")"
5070 NEXT
5080 PRINT#8:PRINT#8,SPC(12);"Sur un TOTAL de";TNB
5090 FOR I=1 TO (62-N):PRINT#8:NEXT
5100 GOSUB 56000
5110 RETURN
54000 '  TRACE DE CERCLE
54010 DEG:PLOT X+R,Y,1+COL
54020 FOR A%=0 TO 360 STEP 10
54030 DRAW R*COS(A%)+X,R*SIN(A%)+Y:NEXT
54040 RETURN
56000 'HARD COPY MODE 2 sur EPSON
56010 PRINT #8
56030  PRINT #8,CHR$(27);CHR$(64);CHR$(27);"A";CHR$(8);
56040 PRINT #8,CHR$(13)
56050 PRINT #8,CHR$(15);
56060 FOR COL=49152 TO 49231:I=0
56070 FOR LI=COL TO COL+1920 STEP 80
56080 FOR T=LI TO LI+14336 STEP 2048
56090 I=I+1:Z(I)=PEEK(T):IF I=200 THEN GOSUB 56120
56100 NEXT:NEXT:PRINT #8,CHR$(13):NEXT
56110 RETURN
56120 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
56130 FOR J=200 TO 151 STEP -1:P=Z(J)
56140 PRINT #8,CHR$(P);CHR$(0);:NEXT
56150 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
56160 FOR J=150 TO 101 STEP -1:P=Z(J)
56170 PRINT #8,CHR$(P);CHR$(0);:NEXT
56180 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
56190 FOR J=100 TO 51 STEP -1:P=Z(J)
56200 PRINT #8,CHR$(P);CHR$(0);:NEXT
56210 PRINT #8,CHR$(27);"K";CHR$(100);CHR$(0);
56220 FOR J=50 TO 1 STEP -1:P=Z(J)
56230 PRINT #8,CHR$(P);CHR$(0);:NEXT
56240 RETURN
65535 ' FIN DE LISTING
```

2

VIE PRATIQUE

Eric GROUSSARD

Ce programme compte trois parties :

BIORYTHME

Entrez votre date de naissance et la date du jour ; vous obtiendrez votre position sur les cycles intellectuel (33 jours), émotionnel (28 jours) et physique (23 jours) : haut, bas ou critique. Les droites verticales en pointillés indiquent la portion des 5, 10, 15, 20, 25e et derniers jours du mois, et la droite, en trait plein, le jour demandé.

ZODIAQUE

Entrez votre date de naissance, et vous saurez votre signe du zodiaque chinois (1 signe par année) et votre signe du zodiaque français avec les décans (attention : les limites des différents signes peuvent varier selon les sources).

CALENDRIER PERPETUEL

Entrez votre date : vous obtiendrez le jour de la semaine, le rang du jour dans l'année, le nombre de jours restants et le calendrier du mois. Valable pour les années 0 à 5000 (donc entrez 1985 et non 85) et tenant compte du passage du calendrier julien au calendrier grégorien. A la question "suite ?", répondre par 0, 1 ou 2 selon les explications du sommaire.

```
1 REM VIE PRATIQUE
2 REM COPYRIGHT ERIC GROUSSARD
10 MODE 1:BORDER 1:INK 0,1:INK 1,24:INK 2,20:INK 3,6
```

```
20 PRINT "                          FIN :0"
30 PRINT
40 PRINT "                   BIORYTHMES :1"
50 PRINT
60 PRINT "SUITE :0 SOMMAIRE"
70 PRINT "       1 AUTRE PERSONNE"
80 PRINT "       2 AUTRE DATE"
90 PRINT
100 PRINT "          SIGNES DU ZODIAQUE :2"
110 PRINT
120 PRINT "SUITE :0 SOMMAIRE"
130 PRINT "       1 AUTRE PERSONNE"
140 PRINT
150 PRINT "        CALENDRIER PERPETUEL :3"
160 PRINT
170 PRINT "SUITE :0 SOMMAIRE"
180 PRINT "       1 AUTRE DATE"
190 PRINT
200 INPUT S
210 CLS
220 IF S=0 THEN END
230 ON S GOTO 270,1290,1940
240 REM-------------------------------
250 REM BIORYTHMES
260 REM-------------------------------
270 PRINT "DATE DE NAISSANCE :"
280 INPUT "J=";J1
290 INPUT "M=";M1
300 INPUT "A=";A1
310 IF S=2 THEN 1290
320 PRINT "DATE ACTUELLE :"
330 INPUT "J=";J2
340 INPUT "M=";M2
350 INPUT "A=";A2
360 CLS
370 PRINT "DATE DE NAISSANCE :";J1;
380 LOCATE 23,1:PRINT "/";M1;
390 LOCATE 27,1:PRINT "/";A1
400 PRINT "DATE ACTUELLE     :";J2;
410 LOCATE 23,2:PRINT "/";M2;
420 LOCATE 27,2:PRINT "/";A2
```

```
430 V=J2
440 ON M1 GOSUB 2360,2370,2380.2390.2400,2410,2420,2430.2440,2450.2460,24
70
450 J1=J1+W
460 ON M2 GOSUB 2360,2370,2380,2390,2400,2410.2420.2430.2440,2450.2460,24
70
470 J2=J2+W
480 IF A2<100 THEN A3=1900+A2 ELSE A3=A2
490 A=INT(365.25*(A2-A1))+J2-J1+1
500 IF A2/4=INT(A2/4) THEN 510 ELSE 540
510 IF J2>59 THEN 520 ELSE 540
520 IF M2>2 THEN A=A+1
530 IF A1<2000 AND A2>1999 THEN A=A-1
540 PRINT "AGE=";A;"JOURS"
550 X=A MOD 33:Y=A MOD 28:Z=A MOD 23
560 PRINT "BIORYTHME INTELLECTUEL=";X;
570 LOCATE 28.4
580 IF X>1 THEN 590 ELSE 620
590 IF X<16 THEN PRINT ":HAUT":GOTO 630
600 IF X>17 THEN 610 ELSE 620
610 IF X<32 THEN PRINT ":BAS":GOTO 630
620 PRINT ":CRITIQUE"
630 PEN 3:PRINT "BIORYTHME EMOTIONNEL  =";Y:
640 LOCATE 28,5
650 IF Y>1 THEN 660 ELSE 690
660 IF Y<13 THEN PRINT ":HAUT":GOTO 700
670 IF Y>15 THEN 680 ELSE 690
680 IF Y<27 THEN PRINT ":BAS":GOTO 700
690 PRINT ":CRITIQUE"
700 PEN 2:PRINT "BIORYTHME PHYSIQUE    =";Z;
710 LOCATE 28,6
720 IF Z>1 THEN 730 ELSE 760
730 IF Z<11 THEN PRINT ":HAUT":GOTO 770
740 IF Z>12 THEN 750 ELSE 760
750 IF Z<22 THEN PRINT ":BAS":GOTO 770
760 PRINT ":CRITIQUE"
770 D=(X-V+34)MOD 33:E=(Y-V+29)MOD 28:  F=(Z-V+24)MOD 23
780 IF M2=2 THEN 790 ELSE 810
790 IF A2/4=INT(A2/4) THEN P=29:GOTO 830
800 P=28:GOTO 830
810 IF M2=4 OR M2=6 OR M2=9 OR M2=11 THEN P=30:GOTO 830
820 P=31
```

```
830 C=1
840 IF D<8 THEN I=16*D+136
850 IF D=8 THEN PLOT 16*C-31,248,1:DRAW 16*C-15,264,1:DRAW 16*C-7,264,1:D
RAW 16*C+1,256,1:GOTO 900
860 IF D>8 AND D<25 THEN I=400-16*D
870 IF D=25 THEN PLOT 16*C-31,16,1:DRAW 16*C-23,8,1:DRAW 16*C-15,8,1:GOTO
 900
880 IF D>25 THEN I=16*D-392
890 IF C=1 THEN PLOT 1,I,1 ELSE DRAW 16*C-15,I,1
900 C=C+1:D=D+1
910 IF D=34 THEN D=1
920 IF C>P THEN 930 ELSE 840
930 C=1
940 IF E<8 THEN I=16*E+136
950 IF E>7 AND E<21 THEN I=360-16*E
960 IF E>20 THEN I=16*E-312
970 IF C=1 THEN PLOT 1,I,3 ELSE DRAW 16*C-15,I,3
980 C=C+1:E=E+1
990 IF E=29 THEN E=1
1000 IF C>P THEN 1010 ELSE 940
1010 C=1
1020 IF F<6 THEN I=16*F+136
1030 IF F>5 AND F<18 THEN I=320-16*F
1040 IF F>17 THEN I=16*F-232
1050 IF C=1 THEN PLOT 1,I,2 ELSE DRAW 16*C-15,I,2
1060 IF F=5 THEN DRAW 16*C-3,228,2
1070 IF F=17 THEN DRAW 16*C-11,44,2:DRAW 16*C+1,56,2
1080 C=C+1:F=F+1
1090 IF F=24 THEN F=1
1100 IF C>P THEN 1110 ELSE 1020
1110 FOR I=1 TO 640
1120 PLOT I,152,1:PLOT I,120,1
1130 NEXT I
1140 FOR I=1 TO 272
1150 IF (I MOD 32)<16 THEN PLOT 65,I,1:PLOT 145,I,1:PLOT 225,I,1:PLOT 305
,I,1:PLOT 385,I,1:PLOT 16*P-15,I,1
1160 PLOT 16*V-15,I,1
1170 NEXT I
1180 PEN 1:INK 1,24
1190 LOCATE 33,13:PRINT "HAUT"
1200 LOCATE 33,17:PRINT "CRITIQUE"
1210 LOCATE 33,21:PRINT "BAS"
```

```
1220 LOCATE 1,7:PRINT :PRINT K$;A3
1230 LOCATE 25,8:INPUT "SUITE 0,1,2";S
1240 CLS
1250 ON S+1 GOTO 10,270,320
1260 REM------------------------------
1270 REM ZODIAQUE CHINOIS
1280 REM------------------------------
1290 REM
1310 INPUT "DATE DE NAISSANCE :J=",J:LOCATE 25,1:INPUT "M=",M:LOCATE 30,1
:INPUT "A=",A
1320 IF A>1900 THEN A=A-1900
1330 IF M=1 THEN K=J
1340 IF M=2 THEN K=J+31
1350 IF M>2 THEN K=60
1360 FOR B=0 TO 85
1370 READ L
1380 IF B=A THEN 1400
1390 NEXT B
1400 H=A MOD 12
1410 IF K>=L THEN H=H+1
1420 IF H=0 THEN C$="DU SANGLIER"
1430 IF H=1 THEN C$="DU RAT"
1440 IF H=2 THEN C$="DU BUFFLE"
1450 IF H=3 THEN C$="DU TIGRE"
1460 IF H=4 THEN C$="DU CHAT"
1470 IF H=5 THEN C$="DU DRAGON"
1480 IF H=6 THEN C$="DU SERPENT"
1490 IF H=7 THEN C$="DU CHEVAL"
1500 IF H=8 THEN C$="DE LA CHEVRE"
1510 IF H=9 THEN C$="DU SINGE"
1520 IF H=10 THEN C$="DU COQ"
1530 IF H=11 THEN C$="DU CHIEN"
1540 LOCATE 11,5:PRINT "ZODIAQUE CHINOIS"
1550 LOCATE 5,8:PRINT "VOUS ETES DU SIGNE ";C$
1560 DATA 31,50,39,29,47,35,25,44,33,22,41,30
1570 DATA 49,37,26,45,34,23,42,32,51,39,28,47
1580 DATA 36,24,44,33,23,41,30,48,37,26,45,35
1590 DATA 24,42,31,50,39,27,46,36,25,44,33,22
1600 DATA 41,29,48,37,27,45,34,24,43,31,49,39
1610 DATA 28,46,36,25,44,33,21,40,30,48,37,27
1620 DATA 46,34,23,42,31,49,38,28,47,36,25,44
1630 DATA 33,51
```

```
1640 REM------------------------------
1650 REM ZODIAQUE FRANCAIS
1660 REM------------------------------
1670 ON M GOTO 1680,1690,1700,1710,1720,1730,1740,1750,1760,1770,1780,179
0
1680 D1=1:D2=11:D3=21:GOTO 1800
1690 D1=1:D2=11:D3=20:GOTO 1800
1700 D1=1:D2=11:D3=21:GOTO 1800
1710 D1=1:D2=11:D3=21:GOTO 1800
1720 D1=2:D2=12:D3=22:GOTO 1800
1730 D1=1:D2=13:D3=22:GOTO 1800
1740 D1=2:D2=13:D3=23:GOTO 1800
1750 D1=3:D2=14:D3=24:GOTO 1800
1760 D1=2:D2=13:D3=24:GOTO 1800
1770 D1=4:D2=14:D3=24:GOTO 1800
1780 D1=2:D2=13:D3=23:GOTO 1800
1790 D1=2:D2=13:D3=22
1800 IF J<D1 THEN N=1:GOTO 1840
1810 IF J<D2 THEN N=2:GOTO 1840
1820 IF J<D3 THEN N=3:GOTO 1840
1830 N=11
1840 N=N+10*M:L=INT(N/10):P=N-10*L:L=L+9:IF L>12 THEN L=L-12
1850 R$=MID$("BELIER    TAUREAU   GEMEAUX   CANCER    LION      VIERGE
 BALANCE   SCORPION  SAGITTAIRECAPRICORNEVERSEAU   POISSONS  ",10*L-9,10)
1860 IF P=1 THEN T$="ER " ELSE T$="EME "
1870 LOCATE 10,14:PRINT "ZODIAQUE FRANCAIS"
1880 LOCATE 5,17:PRINT "VOUS ETES ";R$;" DU";P;T$;"DECAN"
1890 LOCATE 1,25:INPUT "SUITE=0,1";S
1900 CLS:RESTORE:ON S+1 GOTO 10,1310
1910 REM------------------------------
1920 REM CALENDRIER PERPETUEL
1930 REM------------------------------
1940 DIM P(12)
1950 INPUT "DATE :J=",J:INPUT "       M=",M:INPUT "       A=",A
1960 V=J:J1=0:P(1)=31:P(3)=31:P(4)=30:P(5)=31:P(6)=30:P(7)=31:P(8)=31:P(9
)=30:P(10)=31:P(11)=30:P(12)=31
1970 IF A/4=INT(A/4) THEN P(2)=29 ELSE P(2)=28
1980 IF A/400=INT(A/400) AND A>1600 THEN P(2)=28
1990 FOR N=1 TO 12
2000 IF N=M THEN N=P(N):GOTO 2030
2010 J=J+P(N)
2020 NEXT N
```

```
2030 IF V>N THEN 2080
2040 IF A<1582 THEN 2090
2050 IF A>1582 THEN 2100
2060 IF J<=277 THEN 2090
2070 IF J>=288 THEN 2100
2080 LOCATE 9,12:PRINT "CE JOUR N'EXISTE PAS.":GOTO 2340
2090 E=4:GOTO 2110
2100 E=1
2110 FOR I=17 TO 50
2120 IF I/4=INT(I/4) THEN 2140
2130 IF A>I*100 THEN E=E-1 ELSE 2150
2140 NEXT I
2150 IF P(2)=29 THEN E=E-1
2160 B=INT(1.25*A)+J+E:B=B MOD 7
2170 ON M GOSUB 2360,2370,2380,2390,2400,2410,2420,2430,2440,2450,2460,24
70
2180 W=0:IF V=1 THEN T$="ER " ELSE T$=""
2190 IF J=1 THEN S$="ER " ELSE S$="EME "
2200 C$=MID$("DIMANCHELUNDI   MARDI   MERCREDIJEUDI   VENDREDISAMEDI  ",8
*B+1,8)
2210 CLS:LOCATE 5,2:PRINT "LE";V;T$;K$;A;"EST UN ";C$
2220 LOCATE 5,4:PRINT "C'EST LE";J;S$;"JOUR DE L'ANNEE"
2230 LOCATE 5,6:PRINT "IL RESTE";337+P(2)-J;"JOURS"
2240 LOCATE 13,9:PRINT K$;" ";A
2250 LOCATE 10,11:PRINT "L  M  M  J  V  S  D"
2260 IF A=1582 AND M=10 THEN LOCATE 10,13:PRINT "1  2  3  4  15 16 17":LO
CATE 10,15:PRINT "18 19 20 21 22 23 24":LOCATE 10,17:PRINT "25 26 27 28 2
9 30 31":GOTO 2340
2270 F=B-(V MOD 7)+8:F=F MOD 7:Y=13:IF F=0 THEN F=7
2280 FOR I=1 TO N
2290 LOCATE 3*F+6,Y:IF I=V THEN PEN 3 ELSE PEN 1
2300 PRINT I
2310 F=F+1:IF F=8 THEN F=1:Y=Y+2
2320 NEXT I
2330 PEN 1
2340 LOCATE 1,25:INPUT "SUITE 0,1";S
2350 CLS:ON S+1 GOTO 10,1950
2360 W=0:K$="JANVIER":RETURN
2370 W=31:K$="FEVRIER":RETURN
2380 W=59:K$="MARS":RETURN
2390 W=90:K$="AVRIL":RETURN
2400 W=120:K$="MAI":RETURN
```

```
2410 W=151:K$="JUIN":RETURN
2420 W=181:K$="JUILLET":RETURN
2430 W=212:K$="AOUT":RETURN
2440 W=243:K$="SEPTEMBRE":RETURN
2450 W=273:K$="OCTOBRE":RETURN
2460 W=304:K$="NOVEMBRE":RETURN
2470 W=334:K$="DECEMBRE":RETURN
```

LASER'S BYKE

Stéphane CLOIREC

Ce programme est inspiré du film "TRON" de Walt Disney. Il est basé sur le célèbre duel à moto entre le héros et le M.C.P. (comprenez : Master Control Program, le software qui dirige l'ordinateur des jeux).

Entièrement en Basic, il permet de jouer à deux ou tout seul et comprend quatre arènes différentes. On peut utiliser indifféremment le clavier ou les joysticks (pour plus d'explication, tapez le programme et lisez attentivement les instructions).
Il vous faudra tout d'abord rentrer le Loader, le premier petit programme, le sauver sur cassette ; puis tapez le listing principal (les deux programmes peuvent être sauvés sous n'importe quel nom).

STRUCTURE DU JEU

120 :
Test jeu à 1 joueur/Jeu à 2 joueurs.
160-290 :
Routine principale à deux joueurs (test de clavier, de collision, fuel...).
330-500 :
Fin de jeu.
540-630 :
Nouvelle partie.
670-790 :
Routine principale à un joueur (cf. 160-290).
830-880 :
Fin de jeu.
920-1000 :
Nouvelle partie.
1040-1080 :

Place la "Warp-Box" à l'écran.

1120-1160 :
Présentation, instructions.

1680-1880 :
Sous-programmes utilisés par instructions.

1920-2030 :
Menu principal, choix du type de jeu.

2070-2140 :
Les différentes arènes du duel.

2180-2280 :
Initialisation (ink, coordonnées, enveloppes,...).

2320-2430 :
Arène libre.

2470-2540 :
Arène labyrinthe.

2580-2620 :
Arène aléatoire.

2660-2680 :
Arène de la mort.

2720-2940 :
Routines sonores.

Programme de présentation

```
40 MODE 0:INK 0,0:BORDER 0:PEN 1:PAPER 0
50 INK 1,2:LOCATE 2,5:PRINT"*** LASER-BYKE ***"
60 ORIGIN 20,345:DRAWR 600,0,1:DRAWR 0,-37,1:DRAWR -600,0,1:DRAWR 0,37,1
70 INK 2,6:PEN 2:LOCATE 7,11:PRINT CHR$(164);" 1985"
80 INK 3,18:PEN 3:LOCATE 2,14:PRINT"Stephane CLOIREC"
90 ORIGIN 34,169:DRAWR 504,0,3
100 INK 4,8:PEN 4:LOCATE 1,21:PRINT"Chargement en cours."
110 SPEED INK 255,255
120 FOR i=1 TO 500:CALL &BD23:CALL &BD22:NEXT
130 SPEED INK 15,15:ORIGIN 0,0
140 RESTORE 170
150 READ note,dur:IF note=-1 THEN 160 ELSE SOUND 4,note,dur,15:GOTO 150
160 RUN"bykel"
170 DATA 478,50,319,50,358,13,379,13,426,13,239,60,319,60,358,13,379,13,4
26,13,239.60,319,60,358,13,379.13,358.13,426,70,-1,0
```

Programme principal

```
80 '
90 GOSUB 2150:    ' initialisations
100 GOSUB 1100:    ' instructions
```

```
110 GOSUB 1890:  ' menu principal
120 ON op GOTO 160,670
130 '
140 '  routine principale pour jeu a deux joueurs
150 '
160 PLOT xb,yb,1:PLOT xr,yr,2
170 SOUND 1,100,1,7,1,1
180 IF INKEY(bd)=0 THEN xb1=1:vb1=0 ELSE IF INKEY(bg)=0 THEN xb1=-1:vb1=0
190 IF INKEY(bh)=0 THEN xb1=0:vb1=1 ELSE IF INKEY(bb)=0 THEN xb1=0:vb1=-1
200 IF INKEY(rd)=0 THEN xr1=1:vr1=0 ELSE IF INKEY(rg)=0 THEN xr1=-1:vr1=0
210 IF INKEY(rh)=0 THEN xr1=0:vr1=1 ELSE IF INKEY(rb)=0 THEN xr1=0:vr1=-1
220 xb=xb+4*xb1:yb=yb+vb1*2
230 xr=xr+4*xr1:yr=yr+yr1*2
240 fuel=fuel-1:IF fuel=0 THEN GOSUB 2830:GOSUB 2780:GOTO 300
250 PLOT 528,366,7:PRINT USING"###";fuel;
260 bcrash=TEST(xb,yb):rcrash=TEST(xr,yr)
270 IF bcrash=4 OR rcrash=4 THEN 290 ELSE IF bcrash<>0 AND rcrash<>0 THEN
 GOSUB 2910:GOSUB 2830:GOSUB 2780:GOTO 300
280 IF bcrash<>0 OR rcrash<>0 THEN GOSUB 2910:GOSUB 2830:GOSUB 2780:GOTO
410
290 GOTO 160
300 '
310 '   Mort des deux joueurs ou panne d'essence
320 '
330 MODE 1:LOCATE 12,3:PRINT"TRON'S LASER-BYKES":PEN 2:LOCATE 12,4:PRINT"
=================="
340 PEN 3:LOCATE 1,7:PRINT" Champions, vous vous etes bien battus."
350 PRINT:PRINT"   Mais votre duel s'est termine d'une  maniere assez dec
evante. Le M.C.P. va  vous dire ce qu'il en pense...":PEN 1
360 IF fuel=0 THEN PRINT:PRINT" Vous n'avez pas eu assez de fuel pour  me
ner a terme ce combat qui devait etre mortel. Vous ne meritez donc pas qu
'on  vous laisse la vie sauve !!!":GOTO 380
370 PRINT:PRINT" Vous avez ete assez stupides pour vous exterminer mutuel
lement. Votre place n' est pas dans cette arene, mais aux minesde siliciu
m de Soracom-city !!!"
380 PEN 3:LOCATE 1,18:PRINT" Neanmoins, dans sa grande magnanimite, le M.
C.P. vous laisse une chance de vousrattrapper."
390 LOCATE 6,22:PEN 2:PRINT"Ne la laisser pas echapper !!!"
400 GOTO 540
410 '
420 ' Mort d'une des deux laser-bykes
430 '
```

```
440 MODE 1:LOCATE 12,3:PRINT"TRON'S LASER-BYKES":PEN 2:LOCATE 12,4:PRINT"
=================="
450 PEN 3:LOCATE 1,7:PRINT" Champions, vous vous etes bien battus."
460 IF bcrash<>0 THEN coul$="rouge":rscore=rscore+1 ELSE coul$="bleue":bs
core=bscore+1
470 PRINT:PRINT"   Mais le conducteur de la laser-byke   "coul$" s'est re
vele le plus fort. C'est";" donc a lui que le  M.C.P.  accorde une  victo
ire en plus."
480 LOCATE 17,15:PEN 1:PRINT"SCORE":LOCATE 17,16:PEN 2:PRINT"=====":PEN 3
490 LOCATE 10,19:PEN 1:PRINT"Joueur Bleu......";:PEN 2:PRINT bscore
500 LOCATE 10,21:PEN 2:PRINT"Joueur Rouge.....";:PEN 1:PRINT rscore
510 '
520 '   On rejoue ?
530 '
540 LOCATE 1,24:PEN 1:PRINT"    Appuyez sur la touche 'ENTER'...":a$=INKE
Y$:IF a$=""THEN 540
550 IF a$<>CHR$(13) THEN 540
560 MODE 1:FOR n=-15 TO 15:SOUND 4,ABS(n)*10.5,15,0,0,ABS(n):NEXT:LOCATE
12,3:PRINT"TRON'S LASER-BYKES":PEN 2:LOCATE 12,4:PRINT"==================
"
570 PEN 1:LOCATE 7,7:PRINT"Voulez-vous rejouer ?        ( O / N )"
580 a$=INKEY$:a$=UPPER$(a$):IF a$<>"O" AND a$<>"N" THEN 580
590 IF a$="N"THEN 110
600 LOCATE 7,9:PRINT"On change l'arene ?          ( O / N )"
610 a$=INKEY$:a$=UPPER$(a$):IF a$<>"O" AND a$<>"N" THEN 610
620 IF a$="N"THEN GOSUB 2150:GOSUB 2130:GOTO 120
630 GOSUB 2150:GOSUB 2070:GOTO 120
640 '
650 '  routine principale pour jeu a un joueur
660 '
670 EI:AFTER temps GOSUB 1010
680 PLOT xb,yb,1
690 SOUND 1,100,1,7,1,1
700 IF INKEY(bd)=0 THEN xb1=1:yb1=0 ELSE IF INKEY(bg)=0 THEN xb1=-1:yb1=0
710 IF INKEY(bh)=0 THEN xb1=0:yb1=1 ELSE IF INKEY(bb)=0 THEN xb1=0:yb1=-1
720 xb=xb+4*xb1:yb=yb+yb1*2
730 fuel=fuel-1:IF fuel=0 THEN GOSUB 2830:GOSUB 2780:GOTO 830
740 PLOT 528,366,7:PRINT USING"###":fuel;
750 IF xb<=32*x AND xb>32*(x-1) AND yb<=16*y AND yb>16*(y-1)THEN vic=1:GO
SUB 2870:GOSUB 2780:GOTO 830
760 bcrash=TEST(xb,yb)
```

```
770 IF bcrash=4 THEN 680 ELSE IF bcrash<>0 THEN GOSUB 2910:GOSUB 2830:GOS
UB 2780:GOTO 830
780 xm=INT(RND*614)+14:vm=INT(RND*320)+6:PLOT xm,vm,12
790 GOTO 680
800 '
810 '  Fin de jeu
820 '
830 MODE 1:DI:LOCATE 10,3:PEN 2:PRINT"----------------------":LOCATE 10,4
:PRINT"|";:PEN 1:PRINT" TRON'S LASER-BYKES ";:PEN 2:PRINT"|":LOCATE 10,5:
PRINT"----------------------"
840 LOCATE 1,8:PEN 3:PRINT"  Vous avez essaye de vous mesurer avec le Mas
ter Control Program,  mais celui- ci n'est jamais a court de ressources .
"
850 IF fuel=0 THEN LOCATE 1,12:PRINT"  Vous n'avez meme pas eu assez de f
uel pour l'inquieter !!!":temps=temps-100:GOTO 920
860 IF vic=1 THEN LOCATE 1,12:PRINT"Pourtant, vous avez momentanement reu
ssia le mettre en deroute...":PRINT"Vous pouvez, bien-sur, rester sur cet
te victoire.":temps=temps+250:bscore=bscore+1:GOTO 920
870 LOCATE 1,12:PRINT"  Le crash final dont vous avez ete la  victime eta
it inevitable : tot ou tard  le M.C.P. vous aurait vaincu...":temps=temps
-100
880 IF vic=0 THEN mscore=mscore+1
890 '
900 '  On rejoue ?
910 '
920 LOCATE 1,18:PEN 1:PRINT"   Neanmoins, voulez-vous refaire une      un
e partie contre moi ?  ( O / N )"
930 a$=INKEY$:a$=UPPER$(a$):IF a$<>"O" AND a$<>"N" THEN 930
940 IF a$="N"THEN GOTO 110
950 LOCATE 15,21:PEN 2:PRINT"HA! HA! HA! ...":FOR i=1 TO 500:NEXT
960 LOCATE 4,23:PEN 1:PRINT"On change l'arene ?      ( O / N )"
970 a$=INKEY$:a$=UPPER$(a$):IF a$<>"O" AND a$<>"N" THEN 970
980 GOSUB 2150
990 IF a$="N" THEN GOSUB 2130:GOTO 120
1000 GOSUB 2070:GOTO 120
1010 '
1020 ' place le but a atteindre
1030 '
1040 TAGOFF:x=INT (RND*18)+2:y=INT (RND*17)+7:LOCATE x,y:PEN 13:PRINT CHR
$(181): TAG
1050 y=26-y
1060 GOSUB 2760
```

```
1070 DI
1080 RETURN
1090 '
1100 ' instructions
1110 '
1120 MODE 1:BORDER 0
1130 LOCATE 10,1:PEN 2
1140 PRINT"----------------------":GOSUB 2800
1150 LOCATE 10,2:PRINT"!":GOSUB 2800
1160 LOCATE 10,3:PRINT"----------------------":GOSUB 2800
1170 PEN 1:CH$="TRON'S LASER-BYKES"
1180 SOUND 7,60,150,15,5,0
1190 FOR j=1 TO 18
1200 LOCATE 41-j,2:PRINT LEFT$(ch$,j)
1210 NEXT
1220 FOR j=22 TO 12 STEP -1
1230 LOCATE j,2:PRINT ch$
1240 NEXT j
1250 FOR j=40 TO 30 STEP -1
1260 LOCATE j,2:PRINT" "
1270 NEXT j
1280 LOCATE 31,2:PEN 2:PRINT"!":GOSUB 2800
1290 ch$="Instructions":GOSUB 1680
1300 GOSUB 1750
1310 PEN 3:LOCATE 4,8:PRINT"Bienvenue a vous, Guerriers du Futur!";"Vous
venez de penetrer dans le monde duret impitoyable des jeux videos, ou seu
lsles plus forts, les plus rapides peuventesperer survivre quelque temps.
.."
1320 LOCATE 4,14:PRINT"Vous avez ete choisis pour participer";"a ce duel
par le  Master Control Program(M.C.P. pour les intimes)Le combat ne seter
minera que par la mort d' un des deuxcombattants."
1330 PEN 2:LOCATE 4,20:PRINT"Inutile de fuir:";:PEN 3:PRINT"Une fois le p
rogramme";"engage, le duel est inevitable.Vos sortssont maintenant lies a
 celui du M.C.P."
1340 GOSUB 1820
1350 GOSUB 2720
1360 ch$="But du Duel":GOSUB 1680
1370 GOSUB 1740
1380 PEN 3
1390 LOCATE 1,8:PRINT"    Vous allez etre places aux commandesde nouveaux
 engins-tueur:les LASER-BYKESCe sont, en quelque sorte, des motos quilais
sent derriere elles un sillage laserdont le contact est mortel."
```

```
1400 PRINT:PRINT"    Votre but sera , bien evidemment, defaire 'crasher'
 l'autre conducteur sanspour autant en faire de meme !"
1410 PRINT:PRINT"    De plus, pour etre en accord avec ladecision du Haut
 Commissaire des Jeux,leM.C.P.a limite la consommation de fuel a999 Megac
redits au depart."
1420 GOSUB 1820
1430 GOSUB 2720
1440 ch$="Variantes":GOSUB 1680:GOSUB 1740
1450 PEN 3:LOCATE 1,8:PRINT"    Vous pouvez, bien-entendu, combattrea deu
x joueurs , mais vous avez aussi lapossibilite de vous re-entrainer  cont
rele M.C.P."
1460 LOCATE 1,13:PRINT"    - Dans le Duel a deux, chacun dirigeune Laser-
Byke."
1470 PRINT:PRINT"    - Contre le M.C.P. : Celui-ci envoiedes mines  qu'il
 faut a tout prix eviterjusqu'a l'apparition du "::PEN 2:PRINT"'WARP-BOX'
"::PEN 3:PRINT" , qui";"permet,si on l'atteint, de s'echapper del'arene."
;
1480 PRINT" (Vous utiliserez la Laser-Byke Bleue)"
1490 GOSUB 1820
1500 GOSUB 2720
1510 ch$="Commandes":GOSUB 1680:GOSUB 1740
1520 PEN 3:LOCATE 1,8
1530 PRINT"  Voici les commandes dirigeant les deuxLaser-Bykes :"
1540 PRINT:PEN 1:PRINT"Laser-Byke Bleue"::PEN 2:PRINT"        Laser-Byke
Rouge"
1550 PEN 3:PRINT"       A                         4
                          !                        !                !

                         !"
1560 PRINT"   X --0-- C                    2 --0-- 3"
1570 PRINT"         !                        !                !
         !                                                   Z

              1"
1580 GOSUB 1820
1590 GOSUB 2720
1600 ch$="Arenes de Combat":GOSUB 1680:GOSUB 1740
1610 PEN 3:LOCATE 1,8:PRINT"     Enfin, le Duel peut se derouler surl'une
 des quatre arenes , de difficultesdifferentes, que possede le M.P.C."
1620 LOCATE 1,13:PEN 1:PRINT"     Rappelez-vous toujours que TRON luimeme
 a fini par etre vaincu par le M.P.C"
1630 PEN 3:PRINT"Aussi,n'esperez pas de victoires faciles"
1640 LOCATE 8,19:PEN 2:PRINT"Neanmoins, Bonne Chance..."
1650 GOSUB 1820
```

```
1660 RETURN
1670 '
1680 PEN 1
1690 FOR i=1 TO LEN(ch$):LOCATE i,5
1700 PRINT MID$(ch$,i,1):SOUND 2,INT(RND*957),5,15
1710 FOR at=1 TO 50:NEXT at
1720 NEXT i
1730 RETURN
1740 '
1750 PEN 2
1760 FOR i=1 TO LEN(ch$):LOCATE i,6
1770 PRINT "=":SOUND 2,INT(RND*957),5,15
1780 FOR at=1 TO 50:NEXT at
1790 NEXT i
1800 RETURN
1810 '
1820 PEN 1:ch$="[Appuvez sur une touche]"
1830 FOR i=1 TO LEN(ch$):LOCATE i+8,24
1840 PRINT MID$(ch$,i,1):SOUND 2,INT(RND*957),5,15
1850 FOR at=1 TO 50:NEXT at
1860 NEXT i
1870 CALL &BB18
1880 RETURN
1890 '
1900 '   Menu principal
1910 '
1920 MODE 1
1930 GOSUB 2150
1940 PEN 1:PAPER 0:BORDER 0
1950 bscore=0:rscore=0:mscore=0:temps=1000
1960 FOR n=-15 TO 15:SOUND 4,ABS(n)*10,5,15,0,0,ABS(n):NEXT
1970 LOCATE 10,3:PEN 2:PRINT"----------------------":LOCATE 10,4:PRINT"!"
;:PEN 1:PRINT" TRON'S LASER-BYKES ";:PEN 2:PRINT"!":LOCATE 10,5:PRINT"---
-------------------"
1980 LOCATE 8,9:PEN 3:PRINT"1   Joueur contre Joueur":LOCATE 8,11:PRINT"2
   Joueur contre le M.C.P.":LOCATE 8,13:PRINT"3   Arret du jeu"
1990 PEN 1:LOCATE 1,16:PRINT"Choisissez votre option: (1,2 ou 3) :":op=VA
L(INKEY$):IF op<1 OR op>3 THEN LOCATE 38,16:PRINT "  ":GOTO 1990
2000 IF op=3 THEN PEN 1:MODE 2:END
2010 LOCATE 4,21:PEN 2:PRINT"Clavier ou Joystick ?  ( C / J ) :":rep$=UPP
ER$(INKEY$):IF REP$<>"C"AND REP$<>"J"THEN LOCATE 38,21:PRINT"  ":GOTO 201
0
```

```
2020 IF REP$="C"THEN rd=5:rg=14:rh=20:rb=13:bd=62:bg=63:bh=69:bb=71:GOTO
2070
2030 bd=75:bg=74:bh=72:bb=73:rd=51:rg=50:rh=48:rb=49
2040 '
2050 '    Choix de l' arene de combat
2060 '
2070 CLS:FOR n=-15 TO 15:SOUND 4,ABS(n)*10,5,15,0,0,ABS(n):NEXT
2080 LOCATE 10,3:PEN 2:PRINT"----------------------":LOCATE 10,4:PRINT"¦"
;:PEN 1:PRINT" TRON'S LASER-BYKES ";:PEN.2:PRINT"¦":LOCATE 10,5:PRINT"---
-------------------"
2090 LOCATE 1,8:PEN 1:PRINT "Les differentes Arenes de Combat :"
2100 LOCATE 1,9:       PRINT "=================================="
2110 LOCATE 3,12:PEN 3:PRINT"1-  Arene ouverte":LOCATE 3,14:PRINT"2-  Are
ne-labyrinthe":LOCATE 3,16:PRINT"3-  Arene aleatoire":LOCATE 3,18:PRINT"4
-  Arene de la Mort"
2120 PEN 1:LOCATE 1,21:PRINT"Choisissez votre option (1,2,3 ou 4) :":lab=
VAL(INKEY$):IF lab<1 OR lab>4 THEN LOCATE 39,21:PRINT"  ":GOTO 2120
2130 ON lab GOSUB 2290,2440,2550,2630
2140 PEN 1:RETURN
2150 '
2160 ' initialisation
2170 '
2180 INK 0,0:INK 1,2:INK 2,6:INK 3,18:INK 4,13:INK 5,11:INK 6,15:INK 7,24
:INK 8,2:INK 9,6:INK 10,8:INK 11,4:INK 12,20:INK 13,19,16
2190 xb=270:yb=20:xr=370:yr=310
2200 xr1=0:xb1=0:yr1=-1:yb1=1
2210 fuel=999:vic=0
2220 ENV 1,9,1,1,1,0,5,9,-1,2:ENT 1,5,1,1,10,-1,1,10,1,1,10,-1,1,5,1,1
2230 ENV 2,15,-1,13
2240 ENV 3,15,-1,30:ENT -3,10,10,1,10,-12,1
2250 ENV 4,15,-1,4
2260 ENV 5,30,2,5
2270 RETURN
2280 ENV 6,15,-1,1:ENV 5,15,-1,10:ENT 1,30,10,1
2290 '
2300 '   Arene ouverte
2310 '
2320 MODE 0
2330 DRAWR 0,330,4:DRAWR 639,0,4:DRAWR 0,-330,4:DRAWR -639,0,4
2340 PLOT 8,4,5:DRAWR 0,322,5:DRAWR 622,0,5:DRAWR 0,-322,5:DRAWR -622,0,5
2350 TAG:PLOT 0,399:DRAWR 639,0:DRAWR 0,-50:DRAWR -639,0:DRAWR 0,50:PLOT
0,374:DRAWR 639,0:PLOT 304,374:DRAWR 0,25
```

```
2360 PLOT 8,392,8:PRINT"BLEU...";:PLOT 316,392,9:PRINT"ROUGE...";
2370 PLOT 8,366,6:PRINT"FUEL RESTANT -->";
2380 ON pp GOTO 2390,2410
2390 PLOT 8,392,8:PRINT"BLEU...";:PLOT 316,392,9:PRINT"ROUGE...";
2400 PLOT 234,392,9:PRINT USING"##";bscore;:PLOT 574,392,8:PRINT USING"##
";rscore;:GOTO 2430
2410 PLOT 234,392,9:PRINT USING"##";bscore;:PLOT 574,392,8:PRINT USING"##
";mscore;
2420 PLOT 8,392,8:PRINT"BLEU...";:PLOT 316,392,9:PRINT"M.C.P...";
2430 RETURN
2440 '
2450 '   Arene-labyrinthe
2460 '
2470 GOSUB 2290
2480 PEN 11
2490 car1$=CHR$(150)+CHR$(156)
2500 car2$=CHR$(147)+CHR$(153)
2510 TAGOFF
2520 FOR i=2 TO 18 STEP 4:FOR j=7 TO 22 STEP 3:LOCATE i,j:PRINT car1$:LOC
ATE i,j+1:PRINT car2$:NEXT j,i
2530 TAG
2540 RETURN
2550 '
2560 '   Arene aleatoire
2570 '
2580 GOSUB 2290
2590 PEN 10
2600 car1$=CHR$(238)
2610 TAGOFF:FOR i=1 TO 70:x=INT(RND*18)+2:y=INT(RND*17)+7:LOCATE x,y:PRIN
T car1$:NEXT i:TAG
2620 RETURN
2630 '
2640 '   Arene de la Mort
2650 '
2660 GOSUB 2290
2670 FOR I=1 TO 20:x=INT(RND*590)+20:a=INT(RND*590)+20:y=INT(RND*295)+20:
b=INT(RND*295)+20:PLOT x,y,12:DRAW a,b,12:NEXT i
2675 TAGOFF:FOR i=1 TO 19:LOCATE 10,i+5:PRINT"  ":NEXT:TAG
2680 RETURN
2689 RETURN
2690 '
2700 ' sons
```

```
2710 '
2720 MODE 1
2730 FOR n=-15 TO 15:SOUND 4,ABS(n)*10,5,15,0,0,ABS(n):NEXT
2740 LOCATE 10,1:PEN 2:PRINT"----------------------":LOCATE 10,2:PRINT"|";:PEN 1:PRINT" TRON'S LASER-BYKES ";:PEN 2:PRINT"|":LOCATE 10,3:PRINT"----------------------"
2750 RETURN
2760 FOR n=1 TO 20:SOUND 4,300+INT(RND*500),8:NEXT
2770 RETURN
2780 FOR a=1 TO 25:CALL &BC40:SOUND 135,(a+10)*4,0,15,3,3:NEXT
2790 RETURN
2800 SOUND 7,0,60,15,4,0,1
2810 FOR a=1 TO 400:NEXT
2820 RETURN
2830 RESTORE 2850
2840 READ note,dur:IF note=-1 THEN 2860 ELSE SOUND 4,note,dur,0,2:GOTO 2840
2850 DATA 1911,120,1911,80,1911,40,1911,120,1607,80,1703,40,1703,80,1911,40,1911,80,2025,40,1911,120,-1,0
2860 FOR a=1 TO 2800:NEXT:RETURN
2870 RESTORE 2890
2880 READ note:IF note=-1 THEN 2900 ELSE SOUND 4,note,20,15,1:GOTO 2880
2890 DATA 60,53,47,45,60,0,45,47,45,40,53,0,53,47,45,36,40,40,45,45,47,53,47,60,-1
2900 FOR a=1 TO 1000:NEXT:RETURN
2910 SOUND 7,0,195,15,2,0,15
2920 FOR x=1 TO 15:INK 0,26:FOR a=1 TO 10:NEXT:INK 0,0:FOR a=1 TO 40:NEXT a,x
2930 FOR a=1 TO 800:NEXT
2940 RETURN
```

2

LE CHOIX D'UNE IMPRIMANTE

Michel ARCHAMBAULT

C'est un outil quasi indispensable pour programmer en Basic dès que l'on dépasse le stade du débutant. Une imprimante constitue aussi la finalité de nombreux logiciels utilitaires, exemples courrier (très supérieur à une bonne machine à écrire), éditions de tableaux, de fichiers, etc.

C'est un investissement lourd, mais de *longue durée ;* une imprimante ne se démode pas comme un micro-ordinateur. Donc, deux raisons pour ne pas se tromper : mieux vaut économiser encore quelques mois, plutôt que d'acquérir un modèle médiocre, générateur d'ennuis de tous genres.

LES TROIS TECHNOLOGIES D'ECRITURE

Les thermiques

Elles utilisent un papier spécial qui bleuit à la température. C'est très silencieux, assez rapide, bon marché (la machine, pas le papier), mais la qualité d'écriture laisse fort à désirer ; et ce n'est pas photocopiable. A déconseiller en micro-informatique domestique.

A stylos-billes

Généralement à quatre couleurs (quatre stylos spéciaux montés sur barillet). Peu bruyantes, belle écriture mais d'une lenteur désespérante. Deux avantages : le prix et la fonction "table traçante", c'est-à-dire que grâce à certains ordres spécifiques à la machine (équivalant à PLOT et DRAW), on peut dessiner, et en plusieurs couleurs. A ne pas confondre avec la "copie d'écran" qui très souvent est impossible.

A aiguilles

Ou "matricielles". Jadis hors de prix (d'où l'ancienne vogue des thermiques), elles sont aujourd'hui à peine plus chères que les autres. C'est

la technologie que nous conseillons : très rapides (100 à 200 caractères par seconde), très belle écriture (sauf certains modèles bon marché), papier ordinaire, en bandes perforées standard ou en feuille à feuille ; au moins quatre tailles de caractères, plus les mêmes en italique ou en écriture "renforcée", possibilité de copie d'écran graphique. Bref, des bêtes à tout faire, et en plus de cela, elles sont d'une surprenante robustesse mécanique (sauf une certaine marque). Deux défauts : aussi bruyantes qu'une machine à écrire — une seule couleur d'encre, le noir.

Les autres
Pour usages professionnels très spécifiques, à savoir les imprimantes à "marguerites" (analogues aux machines à écrire à boule IBM). Super-qualité courrier, mais très lentes (≃ 15 CPS = caractères par seconde) et caractères limités à ceux de la marguerite installée. Plus chères que les matricielles, leurs prix ont tendance à baisser. Réservées exclusivement au traitement de texte classe PRO.
Les imprimantes à jet d'encre, pour des tables traçantes professionnelles.
Les imprimantes au Laser. Tout pour plaire : qualité, rapidité (environ une page en six secondes...), mais encore hors de prix, même en pro. Leur principe rappelle celui de la photocopie lancée par XEROX.

Combien faut-il compter pour une *bonne* imprimante à aiguilles ? Entre 3000 et 4500 francs TTC. Certes, c'est cher, mais c'est pour longtemps. On change de micro-ordinateur, mais on conserve son imprimante. La preuve, vous n'en voyez pratiquement pas en occasion ; sauf des modèles bas de gamme dont on veut se débarasser.

COMMENT ÇA MARCHE ?

L'électronique d'une imprimante rappelle celle d'un micro-ordinateur : elle a sa ROM (son langage) et une petite RAM appelée "buffer" ou "tampon". Supposons que le micro lui envoie la lettre A majuscule, code ASCII 65, donc envoi du nombre 65 en binaire sur un octet. Les huit bits en même temps ; il y a donc, dans le câble de liaison, au moins huit fils numérotés "D0" à "D7" (voir le schéma de la page A5.2 du manuel AMSTRAD), auxquels s'ajoutent un fil de masse plus deux autres de "conversation" appelés STROBE et BUSY. Dialogue du genre : "Es-tu prête à recevoir ?" "Non, je n'ai pas fini d'imprimer... Vas-y ! ... Stop ! mon buffer est plein", etc.
Nous venons de décrire le mode de transmission "parallèle" (les 8 bits ensemble) aussi appelé standard "CENTRONICS" ; nom du fabricant d'imprimantes qui l'a lancé. C'est celui utilisé par AMSTRAD et la grande majorité des autres micro-ordinateurs domestiques ou professionnels.

Toutes les prises côté imprimante respectent cette norme, mais pas côté micros, d'où des câbles spécifiques à chaque marque de micros, ce qui est complètement stupide et regrettable.
Il existe aussi la transmission en "série" ou "RS 232 C" (câble à deux conducteurs). C'est (heureusement !) beaucoup plus rare et très mal normalisé. N'en parlons pas.
Donc, notre imprimante reçoit pour "A" le nombre 65 : sa ROM va commander les aiguilles de la tête d'impression afin qu'elles impriment tel dessin, ici le A. Quelle forme de A ? La forme banale, courante, prise par défaut dès la mise sous tension de l'imprimante. C'est la taille "PICA", 80 caractères par ligne de 20 cm de long. Voulez-vous d'autres tailles de caractères ? Facile : envoyons *d'abord* à l'imprimante des ordres codés qu'elle va conserver en RAM dans son "buffer de consignes".

LES CODES DE CONSIGNES

Ils commencent généralement par CHR$(27). Exemple :

PRINT #8,CHR$(27);"4"; "Vivement Dimanche"

va provoquer la forme dite en "italique". Les phrases suivantes seront aussi en italique. Pour annuler cette consigne, trois méthodes :
— un code de contre-ordre, ici CHR$(27);"5" ;
— un code qui vide tout le buffer : CHR$(27);"   " ;
— éteindre et rallumer l'imprimante.
Au point noir (si j'ose dire...), toutes les imprimantes n'ont pas le même langage. Celui que nous illustrons est celui des imprimantes Epson, langage heureusement identique sur bien d'autres marques. C'est une sorte de "standard" officieux, un peu comme le Basic de l'AMSTRAD qui est considéré comme "standard" parce que très ressemblant au "Basic MICROSOFT". Dans notre exemple, il y a le listing du programme Basic et ensuite ce que cela donne sur la *plupart* des imprimantes.
Dans ce petit programme de démonstration, nous avons illustré quelques fonctions parmi les 53 dont on dispose sur l'EPSON RX80.

```
10 'DEMONSTRATION D'IMPRESSIONS SUR EPSON RX-80
20 PRINT#8,CHR$(27)"@";: 'Nettoie le Buffer,sans sa
ut de ligne
30 PRINT#8,"Ecriture NORMALE forme PICA:80 CARACTE
RES PAR LIGNE."
40 PRINT#8: 'SAUTE UNE LIGNE BLANCHE
50 PRINT#8,CHR$(27)"W"CHR$(1);"CARACTERES LARGES":
PRINT#8
60 PRINT#8,CHR$(27)"E";"ON RENFORCE L'ECRITURE.":P
RINT#8
70 PRINT#8,CHR$(27)"W"CHR$(0): 'SUPPRIME LE MODE LA
```

```
RGE.PAS DE ; =SAUT D'UNE LIGNE
80 PRINT#8,"L'ECRITURE RESTE RENFORCEE":PRINT#8
90 PRINT#8,CHR$(27)"4";"Aimez-vous l'ITALIQUE ?"
100 PRINT#8,CHR$(27)"5";:'SUPPRIME L'ITALIQUE
110 PRINT#8,CHR$(27)"M";"PLUS PETIT VOICI LA FORME
 DITE ELITE."
120 PRINT#8,CHR$(27)"P"CHR$(27)"F":'SUPPRIME ELITE
 ET RENFORCE
130 PRINT#8,CHR$(15);"LE MODE CONDENSE PERMET DE L
OGER 132 CARACTERES PAR LIGNE.":PRINT#8
140 PRINT#8,CHR$(18);:'SUPPRIME MODE CONDENSE
150 PRINT#8,CHR$(27)"W"CHR$(1);CHR$(27)"M";CHR$(27
)"4";CHR$(27)"E";"COMBINONS LES MODES ITALIQUE,"
160 PRINT#8,"LARGE,ELITE ET RENFORCE.
170 PRINT#8,CHR$(27)"@"
180 PRINT#8,"ON PEUT AUSSI ";CHR$(27)"-"CHR$(1);"S
OULIGNER UN MOT";CHR$(27)"-"CHR$(0);" OU UNE PHRAS
E":PRINT#8
190 PRINT#8,"OU DECALER DES INDICES ET EXPOSANTS:"
200 PRINT#8,"H";CHR$(27)"S"CHR$(1);"2";CHR$(27)"T"
;"O   E=MC";CHR$(27)"S"CHR$(0);"2";CHR$(27)"T":PRI
NT#8
210 PRINT#8,"OU FAIRE DES SURCHARGES: XXXXXX";CHR$
(8);CHR$(8);CHR$(8);"------"
220 PRINT#8:PRINT#8,"CHANGER LA NATIONALITE DE CER
TAINS CODES ASCII:"
230 C$="# $ @ [ ] { } \ ^ ` |  "+CHR$(126)
240 PRINT#8,"POUR USA,FRANCE,ALLEMAGNE,ANGLETERRE,
DANEMARK,SUEDE,ITALIE,ESPAGNE,JAPON ET NORVEGE"
250 PRINT#8,"ON A RESPECTIVEMENT POUR ";C$:PRINT#8
260 FOR I=0 TO 9:PRINT#8,CHR$(27)"R"CHR$(I);C$
270 NEXT:PRINT#8,CHR$(27)"@":PRINT#8
280 PRINT#8,"AVEC LE MODE BIT IMAGE C'EST L'IMAGE
BINAIRE VERTICALE DU NOMBRE RECU QUI EST IMPRIME.V
OICI 80 FOIS 255:"
290 PRINT#8,CHR$(27)"K"CHR$(80);CHR$(0);
300 FOR N=1 TO 80:PRINT#8,CHR$(255);:NEXT
```

```
Ecriture NORMALE forme FICA:80 CARACTERES PAR
LIGNE.
CARACTERES LARGES

ON RENFORCE L'ECRITURE.


L'ECRITURE RESTE RENFORCEE
```

Aimez-vous l'ITALIQUE ?
PLUS PETIT VOIC' LA FORME DITE ELITE.

LE MODE CONDENSE PERMET 'E LOGER 132 CARACTERES PAR LIGNE.

C O M B I N O N S   L E S   M O D E S   I T A L I Q U E
L A R G E , E L I T E   E T   R E N F O R C E .

ON PEUT AUSSI <u>SOULIGNER UN MOT</u> OU UNE PHRASE

OU DECALER DES INDICES ET EXPOSANTS:
H_2O $E=MC^2$

OU FAIRE DES SURCHARGES: XXX***---

CHANGER LA NATIONALITE DE CERTAINS CODES ASCII:
POUR USA,FRANCE,ALLEMAGNE,ANGLETERRE,DANEMARK,
SUEDE,ITALIE,ESRAGNE,JAPON ET NORVEGE
ON A RESPECTIVEMENT POUR # $ @ [ ] { } \ ^ ` |

#	$	@	[	]	{	}	\	^	`	|	~
#	$	à	°	§	é	è	ç	^	`	ù	¨
#	$	§	Ä	Ü	ä	ü	ö	^	`	ö	ß
£	$	@	[	]	{	}	\	^	`	|	~
#	$	@	Æ	Å	æ	å	Ø	^	`	ø	~
#	¤	É	Ä	Å	ä	å	Ö	Ü	é	ö	ü
#	$	@	°	é	à	è	\	^	ù	ò	ì
Pt	$	@	¡	¿	¨	}	Ñ	^	`	ñ	~
#	$	@	[	]	{	}	¥	^	`	|	~
#	¤	É	Æ	Å	æ	å	Ø	Ü	é	ø	ü

AVEC LE MODE BIT IMAGE C'EST L'IMAGE BINAIRE
VERTICALE DU NOMBRE RECU QUI EST IMPRIME.
VOICI 80 FOIS 255:

LA TETE D'IMPRESSION

Une merveille de micromécanique ! Dans le texte mobile, il y a neuf aiguilles disposées les unes au-dessus des autres ; soit neuf pistons métalliques commandés par neuf micro-électro-aimants. Elles viennent frapper un ruban encreur derrière lequel est le papier. La frappe d'un caractère, un A par exemple, se fait en cinq déplacements latéraux de la tête de gauche à droite. Cent caractères par seconde, cela fait donc

500 frappes par seconde ! On reste confondu devant une telle vitesse, une telle précision et aussi une telle robustesse. Je connais des petites imprimantes qui tournent cinq à six heures par jour depuis un an et demi sans toujours la moindre panne... Seul le ruban est changé tous les mois (voire tous les deux mois).

LA QUALITE D'IMPRESSION

Certains modèles bon marché ont moins de neuf aiguilles, et le résultat n'est pas très joli, surtout en minuscules où il n'y a pas de jambages inférieurs. Ainsi, les lettres "f, g, j, p et y" sont "remontées" dans la ligne. C'est le cas de l'imprimante rebaptisée par AMSTRAD et de nombreux modèles SEIKOSHA. Voilà pourquoi elles sont moins chères... Ce n'est absolument pas présentable pour du courrier (avis personnel largement partagé).
Autre point à vérifier : la nature du ruban encreur. Il faut qu'il soit très long, type machine à écrire. S'il est présenté en cartouche, attention au réapprovisionnement pour les marques peu connus, car chaque fabricant a ses modèles de cartouches... Un bon point pour la GEMINI STAR qui utilise une bobine standard. Un zéro pointé pour le modèle étiqueté AMSTRAND dont la (chère) cartouche ne contient qu'une boucle de 50 cm avec passage sur une ampoule d'encre ! Résultats : démarrages avec ruban sec et caractères pâles et une rente en cartouches. Où se trouve alors l'économie ?

LA COMPATIBILITE

Le cas est rare, mais on rencontre parfois des incompatibilités d'humeur entre tel micro et telle imprimante. Par exemple, saut de ligne entre chaque ligne ("LINE FEED"). Souvent, il suffit de modifier la position d'un switch dans l'imprimante, parfois il n'y a pas de solution.
Donc, avant d'acheter une imprimante, *essayez-la* chez le vendeur, avec un AMSTRAD et un câble approprié.

LES CRITERES DE DIVISION

Nous venons de voir les deux principaux, à savoir les jambages inférieurs des minuscules et la nature du ruban encreur. En voici quelques autres importants :
— La possibilité de "clavier" français, qui permet d'imprimer les caractères minuscules accentués. C'est quasi indispensable.
— L'écriture renforcée. C'est très pratique lorsque le ruban demande sa mise à la retraite.

— Friction/Traction. La position ''Friction'' permet de charger des feuilles séparées, à en-tête, par exemple.
— Pour faire du ''Hard Copy'' d'écran (graphiques), il est obligatoire de disposer du BIT IMAGE et du pas d'interligne variable.

A l'inverse, ne vous laissez pas séduire par des caractéristiques en fait peu importantes, à savoir :
— La vitesse : 80 CPS est bien suffisant.
— Les tabulations horizontales, verticales et la marge droite. Même si vous les avez, un bon conseil, ne vous en servez pas ! Programmez des SPC et comptez vos lignes en Basic, c'est beaucoup plus sûr... Pour le nombre de caractères par ligne, utilisez la commande AMSTRAD WIDTH. Après un WIDTH(40), un LIST # 8 fournira un listing identique à celui de l'écran en MODE 1. Attention ! une commande WIDTH reste dans le buffer de l'AMSTRAD, pas dans celui de l'imprimante...

Je pense que vous avez tous les éléments pour décider de cet achat ou non. Un dernier détail : à l'image des téléviseurs et des machines à laver, les fabricants d'imprimantes sont très peu nombreux, mais les ''marques'' sont, elles, très nombreuses.

MATRIX

Michel CHANAUD

MATRIX est un jeu de réflexion où vous jouez contre l'ordinateur. Une grille est remplie de valeurs numériques aléatoires positives ou négatives. Vous pouvez choisir une valeur sur une rangée en déplaçant le curseur à l'aide des flèches horizontales et en pressant la touche COPY. A son tour, le CPC choisira une valeur, mais cette fois-ci dans une colonne.
Les compteurs des deux joueurs accumulent les points obtenus lors de chaque choix. Attention ! Il faut parfois savoir faire des coups "faibles" si l'on veut gagner à plus long terme.

```
10 REM                                     ***************************
*******     *                       M.Chanaud  *      *    M A T R I X
  le Canebal *     *                      SARLAT     *      *************
********************
40 INK 0,13:INK 1,0:INK 2,6:INK 3,24:BORDER 13:MODE 1
50 LOCATE 13,5:PEN 3:FOR I%=1 TO 3:PRINT TAB(13)STRING$(15,CHR$(143)):NEX
T
55 LOCATE 28,5:PEN 1:PRINT CHR$(132):LOCATE 28,6:PEN 1:PRINT CHR$(133):LO
CATE 28,7:PRINT CHR$(133)
56 LOCATE 13,8:PRINT CHR$(130)+STRING$(14,CHR$(131))+CHR$(129)
60 PRINT CHR$(22)+CHR$(1):LOCATE 15,6:PEN 1:PRINT"M A T R I X":PRINT CHR$
(22)+CHR$(0)
80 PEN 1: LOCATE 6,16:PRINT"VOULEZ-VOUS LES REGLES DU JEU ? (O/N)"
90 R$=UPPER$(INKEY$):IF R$="" THEN 90
100 IF LEFT$( R$,1)="O" THEN GOSUB 1130
110 REM ---------------------------------        I N I T I A L I S A T I
O N S          --------------------------------
120 CLS: LOCATE 12,10:PRINT"UN INSTANT S.V.P."
130 DIM B$(8),A$(64),R(8):O$=CHR$(207)+CHR$(207)
```

```
140 RANDOMIZE TIME
150 DATA 15,**,10,09,09,09,08,08,08,07,07,07,06,06,06,05,05,05,04,04,04,0
3,03,03,02,02,02,01,01,01,00,00,00,00,00,00,-1,-1,-1,-2,-2,-2,-3,-3,-3,-4
,-4,-4,-5,-5,-5,-6,-6,-6,-7,-7,-7,-8,-8,-8,-9,-9,-9,-9
160 FOR I%=1 TO 64:READ A$(I%):NEXT
170 FOR V%=1 TO 8:FOR H%=1 TO 8
180 A=INT(RND*64)+1:a%=a:IF A$(A%)="##" THEN 180
190 IF A$(A%)="**" THEN A$(A%)=0$
200 B$(V%)=B$(V%)+A$(A%):A$(A%)="##":IF RIGHT$(B$(V%),2)=0$ THEN CTAB%=H%
*2-1:LTAB%=V%
210 NEXT:NEXT
220 FOR I%=1 TO 8:BB$(I%)=B$(I%):NEXT:CCTAB%=CTAB%:LLTAB%=LTAB%
230 S%=0:T%=0:LP%=0:MO%=2
240 GOSUB 730
250 REM ----------------------------------          MOUVEMENT DU JOUEUR
          ----------------------------------
260 IF B$(LTAB%)=0$+0$+0$+0$+0$+0$+0$+0$ THEN GOTO 930
270 GOSUB 1050
280 IF INKEY(9)=0 AND MID$(B$(LTAB%),CTAB%,2)<>0$ THEN 360
290 DC%=-2*(INKEY(1)=0)+2*(INKEY(8)=0):IF DC%=0 THEN  LOCATE (3*CTAB%+7)/
2,2*LTAB%+2:PEN 2:PRINT  MID$(B$(LTAB%),CTAB%,2):FOR I=1 TO 10:NEXT:LOCAT
E (3*CTAB%+7)/2,2*LTAB%+2:PEN 1:PRINT  MID$(B$(LTAB%),CTAB%,2):FOR I=1 TO
 10:NEXT :GOTO 280
300 CTAB%=CTAB%+DC%
310 IF CTAB%=-1 THEN CTAB%=15
320 IF CTAB%=17 THEN CTAB%=1
330 IF MID$(B$(LTAB%),CTAB%,2)=0$ THEN GOTO 300
340 LOCATE (3*CTAB%+7)/2,2*LTAB%+2:PEN 2:PRINT  MID$(B$(LTAB%),CTAB%,2):F
OR I=1 TO 10:NEXT:LOCATE (3*CTAB%+7)/2,2*LTAB%+2:PEN 1:PRINT  MID$(B$(LTA
B%),CTAB%,2):FOR I=1 TO 10:NEXT
350 FOR I=1 TO 30:NEXT:GOTO 280
360 PEN 1:LP%=VAL(MID$(B$(LTAB%),CTAB%,2))
370 LOCATE (3*CTAB%+7)/2.2*LTAB%+2:PRINT 0$:B$(LTAB%)=LEFT$(B$(LTAB%),CTA
B%-1)+0$+RIGHT$(B$(LTAB%),15-CTAB%)
380 S%=S%+LP%:MO%=1
390 GOSUB 890
400 REM-----------------------------------          JEU DU CPC
          ----------------------------------
410 GOSUB 1080
420 FOR N%=1 TO 8:R(N%)=0:NEXT
430 FOR N%=1 TO 8:V%=-100:IF N%=LTAB% THEN 560
440 IF MID$(B$(N%),CTAB%,2)<>0$ THEN 480
```

```
450 W%=-100
460 IF N%=LTAB% THEN 550
470 GOTO 490
480 W%=VAL(MID$(B$(N%),CTAB%,2))
490 FOR M%=1 TO 15 STEP 2
500 IF M%=CTAB% THEN 540
510 IF MID$(B$(N%),M%,2)=O$ THEN 540
520 IF VAL(MID$(B$(N%),M%,2))<V% THEN 540
530 V%=VAL(MID$(B$(N%),M%,2))
540 NEXT M%
550 R(N%)=W%-V%
560 NEXT N%
570 V%=-100
580 FOR N%=1 TO 8
590 IF MID$(B$(N%),CTAB%,2)=O$ THEN 630
600 IF R(N%)<V% THEN 630
610 V%=R(N%)
620 LTAB%=N%
630 NEXT N%
640 LP%=VAL(MID$(B$(LTAB%),CTAB%,2))
650 T%=T%+LP%
660 LOCATE (3*CTAB%+7)/2,2*LTAB%+2:PRINT O$
670 B$(LTAB%)=LEFT$(B$(LTAB%),CTAB%-1)+O$+RIGHT$(B$(LTAB%),15-CTAB%)
680 LOCATE (3*CTAB%+7)/2,2*LTAB%+2:PRINT O$
690 MO%=2
700 GOSUB 890
710 GOTO 250
720 END
730 REM----------------------------------          TABLEAU DE JEU
          ----------------------------------
740 CLS:LOCATE 11,1:PEN 3:PRINT"M A T R I X":PEN 1
750 LOCATE 4,3:PRINT CHR$(150);:FOR I%=1 TO 7:PRINT CHR$(154)+CHR$(154)+C
HR$(158);:NEXT:PRINT CHR$(154)+CHR$(154)+CHR$(156)
760 L$=CHR$(149)+"  ":K$=CHR$(154)+CHR$(154)+CHR$(159)
770 FOR J%=1 TO 7: PRINT TAB(4)
780 FOR I%=1 TO 8:PRINT L$;:NEXT:PRINT CHR$(149)
790 PRINT TAB(4)CHR$(151);:FOR I%=1 TO 7:PRINT K$;:NEXT:PRINT CHR$(154)+C
HR$(154)+CHR$(157)
800 NEXT J%
810 PRINT TAB(4);:FOR I%=1 TO 8:PRINT L$;:NEXT:PRINT CHR$(149)
820 PRINT TAB(4)CHR$(147);:FOR I%=1 TO 7:PRINT CHR$(154)+CHR$(154)+CHR$(1
55);:NEXT:PRINT CHR$(154)+CHR$(154)+CHR$(153)
```

```
830 FOR Y%=1 TO 8:FOR CP%=0 TO 7
840 X%=2*CP%+1
850 LOCATE X%+CP%+4,2+2*Y%:IF MID$(B$(Y%),X%,2)="15" THEN PEN 3 ELSE PEN
1
860 PRINT MID$(B$(Y%),X%,2)
870 NEXT:NEXT
880 RETURN
890 REM---------------------------------          AFFICHAGE DES POINTS
             ---------------------------------
900 LOCATE 30,9:PRINT"VOUS:"S%:LOCATE 30,11:PRINT"CPC :"T%
910 LOCATE 9,21:PRINT"DERNIER COUP ="LP%:LOCATE 10,23:IF MO%=1 THEN PRINT
"AU CPC DE JOUER" ELSE PRINT "A VOUS DE JOUER"
920 RETURN
930 REM---------------------------------             FIN DE LA PARTIE
             ---------------------------------
940 WINDOW #0,1,40,20,25:CLS:WINDOW #0,1,40,1,25
950 LOCATE 10,21:PEN 3
960 IF S%=T% THEN PRINT"  MATCH NUL"
970 IF S%>T% THEN PRINT"BRAVO...VOUS AVEZ GAGNE."
980 IF S%<T% THEN PRINT"      J'AI GAGNE."
990 PEN 1:LOCATE 5,24:PRINT" VOULEZ-VOUS:":LOCATE 20,23:PRINT"1- LA MEME
PARTIE":LOCATE 20,24:PRINT"2- UNE AUTRE PARTIE":LOCATE 20,25:PRINT"3- ARR
ETER"
1000 R$=INKEY$:IF R$<>"" THEN 1000
1010 R$=INKEY$:IF R$="" THEN 1010
1020 IF R$="3" THEN CLS:END
1030 IF R$="1" THEN CLS:FOR I%=1 TO 8:B$(I%)=BB$(I%):NEXT:LTAB%=LLTAB%:CT
AB%=CCTAB%:GOTO 230
1040 RUN 110
1050 REM---------------------------------         CLIGNOTEMENT DU CURSEU
R            ---------------------------------
1060 LOCATE (3*CTAB%+7)/2,2*LTAB%+2:PEN 2:PRINT O$:FOR I=1 TO 10:NEXT:LOC
ATE (3*CTAB%+7)/2,2*LTAB%+2:PEN 1:PRINT O$:FOR I=1 TO 10:NEXT
1070 R$=INKEY$:IF R$="" THEN 1060 ELSE RETURN
1080 REM---------------------------------          ANALYSE DE LA COLONNE
             ---------------------------------
1090 C$=""
1100 FOR N%=1 TO 8:C$=C$+MID$(B$(N%),CTAB%,2):NEXT
1110 IF C$=O$+O$+O$+O$+O$+O$+O$+O$ THEN 930
1120 RETURN
1130 REM --------------------------------            REGLES DU JEU
             ---------------------------------
```

```
1140 CLS
1150 PRINT:PRINT"UNE GRILLE EST AFFICHEE. CONTENANT DES":PRINT:PRINT"NOMB
RES DISPOSES DE FACON ALEATOIRE."
1160 PRINT:PRINT"LE BUT DU JEU EST D'OBTENIR PLUS DE "
1170 PRINT:PRINT"POINTS QUE L'ORDINATEUR."
1180 PRINT:PRINT"QUAND C'EST A VOUS DE JOUER,VOUS POUVEZ"
1190 PRINT:PRINT"DEPLACER LE CURSEUR HORIZONTALEMENT":PRINT:PRINT"AVEC LE
S FLECHES  "+CHR$(242)+"  ET  "+CHR$(243)+" ET PRENDRE"
1200 PRINT:PRINT"LE NOMBRE CHOISI EN APPUYANT SUR [COPY]."
1210 PRINT"L'ORDINATEUR CHOISIT DANS LA COLONNE."
1220 LOCATE 10,24:PRINT"APPUYEZ SUR UNE TOUCHE"
1230 R$=INKEY$:IF R$="" THEN 1230
1240 RETURN
1250 END
```

2

TOUS À TABLE

Michel ARCHAMBAULT

Le plan de table d'un banquet se doit de respecter les "règles" suivantes : chaque convive doit avoir comme voisins et vis-à-vis une personne de sexe opposé, et aucune de ces trois personnes ne doit être son conjoint ; sauf cas particuliers (jeunes mariés).

Ce fastidieux casse-tête est résolu instantanément par ce programme, mieux, il vous propose plusieurs solutions répondant aux critères ci-dessus : deux colonnes de noms à l'écran vous présentent les personnes en vis-à-vis.
Lorsqu'une des solutions vous plaît, inutile de la recopier, pressez la touche E, et elle est éditée sur imprimante.
Ce programme n'est pas d'une utilisation fréquente (hélas, hélas !...), mais sa conception, uniquement logique et aléatoire (il n'y a pas de RND), constitue un exercice Basic qui, lui, n'a rien de futile !
Si vous n'avez pas l'intention de le recopier au clavier, essayez quand même le module du titre (lignes en 12000) dont le type d'animation pourra vous inspirer pour des programmes personnels.

L'ENTREE DES NOMS

Elle se fait en trois étapes : la liste des couples, celle des filles, celle des garçons. Le nombre total de convives est limité à quarante, ce pour des raisons de présentation à l'écran.

COUPLES
(lignes en 3000)

L'écran de saisie présente "Mr, Mme", vous complétez par le nom. En fin de liste, on tape "Q" (quitter). Le nom tapé est entré en deux

tableaux DIM ; M$(C) pour les maris, et E$(C) pour les épouses. L'adjonction du "Mr" ou du "Mme" est automatique.

FILLES
(lignes en 4000)

Les noms entrés sont mis en DIM F$(F). Le préfixe "Mle" est automatique.

GARÇONS
(lignes en 5000)

Même méthode, mise en DIM G$(G). Le préfixe automatique est constitué par deux astérisques.

En cours de saisie, le total TOT est incrémenté ; s'il atteint quarante, on est prévenu par un bip sonore, et le programme passe à la suite. TOT est bien sûr égal à $(C \times 2) + F + G$.
Dans les trois écrans de saisie, les noms sont autopositionnés sur deux colonnes. En bas de l'écran figurent les décomptes incrémentés, par exemple, "5 Demoiselles **Total = 18". Le passage d'un tableau à l'autre se commande par la lettre "Q".

LA DISPOSITION

En fin de saisie, nous possédons quatre listes, les DIM M$, E$, F$ et G$, avec lesquels le programme va créer une cinquième, D$(D), D signifiant "Disposition". C'est une suite de noms avec alternance masculin-féminin et éloignement des conjoints. C'est là, le "gros morceau" !
Une fois cette suite établie, nous l'afficherons sous forme de boucle fermée autour d'une table longue. En faisant tourner cette boucle d'un cran à chaque fois, nous pourrons changer les vis-à-vis, tout en conservant ses voisins droite-gauche. OK ?
Ces "calculs" concernent le traitement des couples, de loin le plus complexe, puis on enchaîne sur le mixage des listes filles et garçons.

LA DISPOSITION DES COUPLES
(lignes en 6000) (figure 1)

Le traitement est différent selon le nombre de couples, C est pair ou impair. Si C est pair, la variable P = 1.
D'abord, un premier passage en zig-zag : M$(1), E$(2), M$(3), E$(4), etc. (voir figure 1a). Ce premier passage est commun aux nombres de couples pairs et impairs.

Second passage : pour C impair, c'est du zig-zag mais en commençant, cette fois, par l'épouse du couple n° 1 : E$(1), MS$(2), E$(3), M$(4), E$(3), etc. que C soit pair ou impair, on termine toujours par une épouse ; normal puisque l'on débute toujours par le mari M$(1).

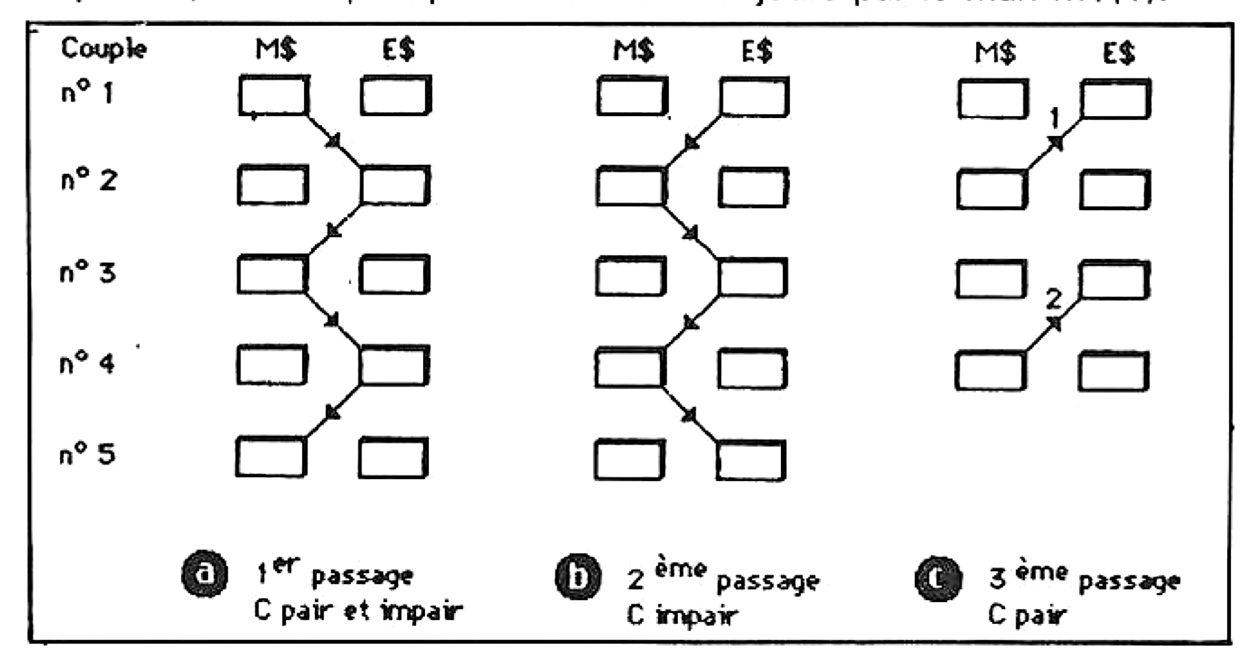


Figure 1

Ce mode de balayage des listes Maris et Epouses assure une dispersion parfaite dans la liste finale D$.

DISPOSITION DES CELIBATAIRES
(lignes en 7000)

La fonction Basic MIN(G,F) fournit le nombre de "couples de célibataires", l'excédent en filles ou en garçons sera conservé tel quel (comment faire autrement). Là, c'est très facile, on prend alternativement un garçon, une fille : G$(1), F$(1), G$(2), F$(2), etc., et on complète par "l'excédent".
Le tableau D$(D) est enfin terminé. On a alors D=TOT.

L'AFFICHAGE
(lignes en 10000)

Les deux colonnes à l'écran figurent les personnes en vis-à-vis. Elles sont séparées par deux caractères graphiques flèche gauche - flèche droite, appartenant au troisième jeu de caractères de l'AMSTRAD, obtenus en faisant précéder de CHR$(1) (ceux obtenus au clavier par CTRL H, CTRL I).
C'est la liste D1(D) qui aurait été pliée en épingle à cheveux, le coude étant en bas de l'écran.

En bas de l'écran figure la légende :
P = Permutation : c'est le changement de vis-à-vis ; peut être répétée un nombre indéfini de fois.
E = Edition (si vous avez une imprimante) : la disposition présentée à l'écran est imprimée. On peut demander une édition (ou plusieurs) de diverses permutations.
Q = Quitter : tout est effacé, on a alors le choix entre l'abandon du programme et une nouvelle saisie de noms.

LA STRUCTURE DU PROGRAMME

L'auteur est un inconditionnel de la programmation "structurée" (ou "modulaire"). Ainsi, tout le scénario du programme est dans les lignes en 1000 (peu nombreuses...). Elles appellent les différents sous-programmes (ou modules) par des GOSUB ; d'où une *souplesse totale* pour des modifications ultérieures, et ce en toute clarté puisque chaque module (ou sous-module) débute par un numéro de ligne multiple de 1000. Une bonne et sage habitude à prendre.
Autres précautions habituelles : en ligne 100, toutes les variables seront des nombres entiers : gain en mémoire (2 octets au lieu de 5...) et boucles FOR-NEXT deux fois plus rapides. D'autre part, on confirme le MODE 1 et l'ORIGIN graphique afin de ne pas "hériter" d'options antérieures qui ne sont pas annulées par NEW ou RUN.
En fin de programme, ligne 1300, on restitue les options couleurs par défaut que l'on avait modifiées. Ce qui s'appelle laisser le micro dans l'état où on l'a trouvé...

LE TITRE ANIME
(lignes en 12000)

Il n'apparaît qu'en fin de chargement. Sur l'écran, se dessine un verre ballon qui se remplit de vin rouge, du Beaujolais bien sûr, car je suis chauvin sur ce point.
Ce grand arc de cercle (le ballon) est tracé par une suite de "cordes de cercle" depuis 135° jusqu'à 405° (lignes 12020 à 12050). C'est très rapide. Le pied est dessiné par des DRAWR (12100-12110).
Le vin : sur un rayon légèrement inférieur, afin de ne pas effacer le verre, on fait PLOT sur un arc de cercle depuis 270° (le fond) jusqu'à 370° ; de là, on trace des traits rouges vers la gauche, de longueurs égales à deux fois le cosinus de l'angle (lignes 12200 à 12230).
On attaque alors le texte :
La chaîne "Disposition des convives" vient se centrer en arrivant par la gauche de l'écran (12310 à 12330). Puis, deux lignes plus bas, la chaîne "autour d'une table" vient se centrer en arrivant par la droite (12340 à 12360).

En bas à gauche de l'écran, le nom de l'auteur et l'année apparaissent caractère par caractère en commençant par le dernier (12400-12410). Deux secondes plus tard, il s'efface de la même manière par PEN = PAPER (12420-12430). Les boucles FOR J = 1 TO... sont des ralentisseurs.
En bas à droite, apparaît alors "Pressez une touche", suivi de CALL &BB06 (ligne 12450), routine qui bloque le programme en attendant un caractère au clavier. C'est plus simple à écrire que WHILE INKEY$ = "":WEND.
NOTE : Toutes ces techniques d'animations simples (et beaucoup d'autres) sont décrites plus en détail dans l'ouvrage du même auteur "MIEUX PROGRAMMER SUR AMSTRAD" (Editions SORACOM). Hé hop ! un peu de pub...

PETITS CONSEILS D'UTILISATION DU PROGRAMME

Il est nécessaire d'avoir un nombre de couples supérieur à deux afin que les conjoints soient séparés.
Pour ne pas séparer un couple, exemple des jeunes mariés à un repas de noce, il suffit de les entrer en tête de liste des célibataires.
Entrez les enfants en dernier, ils seront ainsi regroupés.
Si le nombre total des convives est impair, il y aura forcément quelqu'un sans vis-à-vis. A vous d'effectuer des permutations successives (touche P) jusqu'à trouver la solution viable, par exemple la personne que l'on peut mettre en bout de table.
Suprême avantage de ce programme : si quelqu'un vous reproche d'avoir été mal placé, vous pourrez lui répondre par cet air connu : "Ce n'est pas de ma faute, c'est l'ordinateur... ".

```
10 ' TOUS A TABLE.
20 'Disposition de Convives autour d'une Table.
30 ' AMSTRAD * Michel Archambault * 1985
100 DEFINT A-Z:ORIGIN 0,0:MODE 1
200 GOSUB 12000:'TITRE
1000 ' DEROULEMENT DU PROGRAMME
1010 DIM M$(21),E$(21),F$(37),G$(37),D$(41)
1020 PAPER 0:PEN 2:BORDER 15:CLS
1030 LOCATE 4,10:PRINT "E = ENTREZ LES NOMS DES CONVIVES."
1040 LOCATE 4,13:PRINT "Q = QUITTER LE PROGRAMME."
1050 R$=UPPER$(INKEY$):IF R$="" THEN 1050
1060 IF R$="Q" THEN 1300
1070 IF R$<>"E" THEN PRINT CHR$(7):R$="":GOTO 1030
1100 GOSUB 3000:'ENTREE COUPLES
1110 GOSUB 4000:'ENTREE FILLES
```

```
1120 GOSUB 5000:'ENTREE GARCONS
1130 GOSUB 6000:'CALCULS COUPLES
1140 GOSUB 7000:'CALCULS CELIBATAIRES
1150 GOSUB 10000:'AFFICHAGE
1200 RUN 1000
1300 BORDER 1:PAPER 0:PEN 1:CLS
1310 END
3000 ' ENTREE DES COUPLES
3010 PAPER 1:PEN 0:BORDER 14:CLS:C=1:L=1
3020 LOCATE 5,2:PRINT "20 COUPLES maxi : QUITTER--> Q"
3100 LOCATE 2,3+L:GOSUB 3500:IF FQ=1 THEN 3200
3110 LOCATE 21,3+L:GOSUB 3500:IF FQ=1 THEN 3200
3120 L=L+1:GOTO 3100
3200 RETURN
3500 LINE INPUT "Mr,Mme ",C$:IF UPPER$(C$)="Q" THEN FQ=1:C=C-1:GOTO 3600
3510 C$=LEFT$(C$,12):M$(C)="Mr "+C$:E$(C)="Mme "+C$
3520 TOT=TOT+2:IF TOT=40 OR C=20 THEN PRINT CHR$(7):FOR I=1 TO 4000:NEXT:
FQ=1:GOTO 3600
3530 PEN 3:LOCATE 9,24:PRINT C;"couples ** Total=";TOT:PEN 0
3540 C=C+1
3600 RETURN
4000 ' ENTREE DES DEMOISELLES
4005 IF TOT=40 THEN 4600
4010 PAPER 1:PEN 0:BORDER 7:CLS:F=1:L=1:FQ=0
4020 LOCATE 3,2:PRINT "36 DEMOISELLES maxi : QUITTER--> Q"
4100 LOCATE 2,3+L:GOSUB 4500:IF FQ=1 THEN 4200
4110 LOCATE 21,3+L:GOSUB 4500:IF FQ=1 THEN 4200
4120 L=L+1:GOTO 4100
4200 RETURN
4500 LINE INPUT "Mle ",C$:IF UPPER$(C$)="Q" THEN FQ=1:F=F-1:GOTO 4600
4510 C$=LEFT$(C$,12):F$(F)="Mle "+C$
4520 TOT=TOT+1:IF TOT=40 OR F=36 THEN PRINT CHR$(7):FOR I=1 TO 4000:NEXT:
FQ=1:GOTO 4600
4530 PEN 3:LOCATE 7,24:PRINT F;"Demoiselles ** Total=";TOT:PEN 0
4540 F=F+1
4600 RETURN
5000 ' ENTREE DES GARCONS
5005 IF TOT=40 THEN 5600
5010 PAPER 1:PEN 0:BORDER 1:CLS:G=1:L=1:FQ=0
5020 LOCATE 5,2:PRINT "36 GARCONS maxi : QUITTER--> Q"
5100 LOCATE 2,3+L:GOSUB 5500:IF FQ=1 THEN 5200
5110 LOCATE 21,3+L:GOSUB 5500:IF FQ=1 THEN 5200
```

```
5120 L=L+1:GOTO 5100
5200 RETURN
5500 LINE INPUT "** ",C$:IF UPPER$(C$)="Q" THEN FQ=1:G=G-1:GOTO 5600
5510 C$=LEFT$(C$,12):G$(G)="** "+C$
5520 TOT=TOT+1:IF TOT=40 OR G=36 THEN PRINT CHR$(7):FOR I=1 TO 4000:NEXT:
FQ=1:GOTO 5600
5530 PEN 3:LOCATE 9,24:PRINT G;"Garcons ** Total=";TOT:PEN 0
5540 G=G+1
5600 RETURN
6000 ' DISPOSITION COUPLES
6010 D=0:IF C=0 THEN 6900
6020 IF INT(C/2)=C/2 THEN P=1
6100 'PASSAGE 1
6110 I=1
6120 D=D+1:D$(D)=M$(I)
6130 IF I<C THEN D=D+1:D$(D)=E$(I+1):I=I+2:GOTO 6120
6150 IF P=1 THEN 6300
6200 ' PASSAGE 2   C IMPAIR
6210 I=1
6220 D=D+1:D$(D)=E$(I)
6230 IF I<C THEN D=D+1:D$(D)=M$(I+1):I=I+2:GOTO 6220
6240 GOTO 6900
6300 ' PASSAGE 2  C PAIR
6310 I=2:D=D-1
6320 D=D+1:D$(D)=M$(I)
6330 D=D+1:D$(D)=E$(I-1)
6340 IF I=C THEN 6900
6350 I=I+2:GOTO 6320
6900 RETURN
7000 'DISPOSITION CELIBATAIRES
7010 A=MIN(G,F):IF G+F=0 THEN 7900
7020 FOR I=1 TO A
7030 D=D+1:D$(D)=G$(I)
7040 D=D+1:D$(D)=F$(I)
7050 NEXT
7060 IF G=F THEN 7900
7070 IF G<F THEN 7200
7100 FOR I=F+1 TO G
7110 D=D+1:D$(D)=G$(I)
7120 NEXT
7130 GOTO 7900
7200 FOR I=G+1 TO F
```

```
7210 D=D+1:D$(D)=F$(I)
7220 NEXT
7900 RETURN
10000 'AFFICHAGE
10010 PAPER 2:PEN 0:BORDER 9:CLS:PRINT
10100 IF INT(TOT/2)<TOT/2 THEN TOT=TOT+1
10110 B=TOT/2
10120 FOR I=1 TO B
10130 PRINT " ";D$(I);:PRINT TAB(19);CHR$(1);CHR$(8);CHR$(1);CHR$(9);" ";
D$(TOT+1-I)
10140 NEXT
10150 PEN 3:LOCATE 1,24:PRINT TOT;"P=Permutation; E=Edition; Q=Quitter":P
EN 0:R$=""
10160 R$=UPPER$(INKEY$):IF R$="" THEN 10160
10170 IF R$="Q" THEN 10900
10180 IF R$="E" THEN GOSUB 11000:R$="":GOTO 10160
10190 IF R$<>"P" THEN 10150
10200 'PERMUTATION
10210 D$(0)=D$(1)
10220 FOR I=1 TO TOT-1:D$(I)=D$(I+1)
10230 NEXT
10240 D$(TOT)=D$(0)
10250 R$="":GOTO 10000
10900 RETURN
11000 'EDITION
11010 PRINT #8,CHR$(27);CHR$(64)
11020 FOR I=1 TO B
11030 PRINT #8," ";D$(I);SPC(20-LEN(D$(I)));" :  ";D$(TOT+1-I)
11040 NEXT
11050 PRINT #8
11060 RETURN
12000 'TITRE
12010 MODE 1:PAPER 2:PEN 0:BORDER 9:CLS:R=100
12020 DEG:PLOT 249,271,0
12030 FOR A=135 TO 405 STEP 10
12040 DRAW R*COS(A)+320,R*SIN(A)+200
12050 NEXT
12100 PLOT 310,100:DRAWR 0,-50:DRAWR -60,-10
12110 DRAWR 140,0:DRAWR -60,10:DRAWR 0,50
12200 R=98
12210 FOR A=270 TO 370
```

```
12220 PLOT COS(A)*R+320,SIN(A)*R+200,3
12230 DRAWR -COS(A)*R*2,0:NEXT
12300 PEN 3:H$="DISPOSITION DE CONVIVES":B$="AUTOUR D'UNE TABLE."
12310 A$=SPACE$((40-LEN(H$))/2-1)+H$
12320 FOR I=1 TO LEN(A$):FOR J=1 TO 100:NEXT
12330 LOCATE 1,3:PRINT RIGHT$(A$,I):NEXT:PEN 0
12340 A$=B$+SPACE$((40-LEN(B$))/2)
12350 FOR I=1 TO LEN(A$):FOR J=1 TO 100:NEXT
12360 LOCATE 41-I,6:PRINT LEFT$(A$,I):NEXT
12400 N$="M.Archambault-1985":FOR I=18 TO 1 STEP -1
12410 LOCATE I+1,24:PRINT MID$(N$,I,1):FOR J=1 TO 100:NEXT:NEXT
12420 FOR J=1 TO 4000:NEXT:PEN 2:FOR I=18 TO 1 STEP -1
12430 LOCATE I+1,24:PRINT MID$(N$,I,1):FOR J=1 TO 100:NEXT:NEXT
12450 PEN 3:LOCATE 22,24:PRINT "pressez une touche";:CALL &BB06
12900 RETURN
65535 '------FIN DE LISTING -------
```

2

AWARI

AWARI est un jeu de réflexion d'origine africaine où vous affrontez votre CPC. De toutes les variantes de ce jeu, celle-ci permet au mieux d'exercer ses talents de tacticien, surtout dans le maniement des grosses maisons. A noter : la particularité de pouvoir se déplacer dans la partie, au moyen des flèches horizontales et de reprendre ses coups, permet de progresser plus rapidement dans l'apprentissage de ce grand classique des jeux de stratégie. Les règles du jeu sont incluses dans le programme.

```
10 '*********************************
20 '*                               *
30 '*      A W A R I    M.C.        *
40 '*                               *
50 '*********************************
70 GOSUB 1460 ' presentation et regles
80 :
90 '========== INITIALISATIONS ========
100 K%=1:Q%=14:P%=13:F%=50:D%=12
110 D$=SPACE$(39)
120 DIM B(Q%),T(Q%),Y(Q%),W(Q%),V(6),U(6),AR(300,15)
130 :
140 '=========== DEPART ===============
150 RANDOMIZE TIME
160 ZB=RND(0):ZB=ZB/Q%:ZA=0.25+ZB:ZB=0.25-ZB
170 FOR J%=1 TO D%:B(J%)=4:NEXT
180 GOSUB 1280
190 GOSUB 1140
200 LOCATE 1,24:PRINT"  VOULEZ-VOUS JOUER LE PREMIER ? (O/N)"
210 GOSUB 1410
```

```
220 GOSUB 2710
230 LOCATE 1,24:PRINT D$
240 IF R$="O" THEN COM%=1:GOTO 400
250 IF R$<>"N" THEN 200
260 :
270 '======= BOUCLE PRINCIPALE =========
280 GOSUB 2700 :AR(COUP%,15)=1
290 PEN 2:LOCATE 1,21:PRINT D$;D$;D$;:LOCATE 1,21:PRINT"  JE REFLECHIS ";
:GOSUB 910
300 IF M%<1 THEN 1380
310 LOCATE 1,21:PRINT D$;:LOCATE 1,21:PRINT"  JE JOUE EN";M%:SOUND 7,100,
10,5
320 SCPC%=0: FOR J%=1 TO Q%:T(J%)=B(J%):SCPC%=SCPC%+B(J%):NEXT:GOSUB 710
330 IF SCPC%<>0 THEN 360
340 OK%=0:FOR I%=1 TO 6:IF B(I%)+I%>=7 THEN OK%=1:NEXT
350 IF OK%=0 THEN 1380
360 FOR J%=1 TO Q%:B(J%)=T(J%):NEXT:GOSUB 1150
370 IF B(Q%)<24 THEN INTER%=0:GOTO 400
380 LOCATE 1,21:PEN 3:PRINT"               J'AI GAGNE !";D$:GOTO 570
390 :
400 '======= JEU DU JOUEUR =============
410 IF INTER%=1 THEN 430
420 LOCATE 1,24:PRINT D$:LOCATE 1,24:PEN 2:PRINT"  QUEL EST VOTRE MOUVEME
NT ? (1 - 6) "
430 GOSUB 1410
440 LOCATE 1,24:PRINT D$
450 IF ASC(R$)=242 THEN GOTO 2600
460 IF ASC(R$)=243 THEN GOTO 2730
470 IF R$<"1" OR R$>"6" THEN 400
480 IF INTER%=1 THEN 430
490 GOSUB 2700:AR(COUP%,15)=0
500 M%=VAL(R$)
510 FOR J%=1 TO Q%:T(J%)=B(J%):NEXT
520 GOSUB 710:IF M%<0 THEN 650
530 FOR J%=1 TO Q%:B(J%)=T(J%):NEXT
540 MN%=MN%+1:GOSUB 1140
550 IF B(P%)<24 THEN 270
560 LOCATE 1,21:PEN 3:PRINT"          BRAVO ! VOUS AVEZ GAGNE.";D$
570 PEN 2:LOCATE 1,24:PRINT"  VOULEZ-VOUS ANALYSER CETTE PARTIE ?"
580 GOSUB 1410
590 IF R$="O" THEN COUP%=0:LOCATE 1,21:PRINT"               "+CHR$(242)+"
ou JEU ou "+CHR$(243)+D$:GOTO 2730
```

```
600 LOCATE 1,24:PRINT"VOULEZ-VOUS JOUER UNE AUTRE PARTIE (O/N)"
610 GOSUB 1410:IF R$="O" THEN 140
620 IF R$="N" THEN CLS:LOCATE 15,15:PEN 3:PRINT"AU REVOIR":CALL &BB18:END
630 GOTO 610
640 :
650 '======== COUP ILLEGAL =============
660 PEN 3:LOCATE 1,24 :FOR i=1 TO 15:SOUND 7,50,2,12:FOR j=1 TO 20:NEXT j
,i
670 IF M%=-1 THEN PRINT"     CETTE CASE EST VIDE."
680 IF M%=-2 THEN PRINT" IL FAUT LAISSER DU JEU A L'ADVERSAIRE."
690 FOR J=1 TO 1000:NEXT:PEN 1:COUP%=COUP%-1:GOTO 400
700 :
710 '======== CASE VIDE ?? ============
720 IF T(M%)=0 THEN M%=-1:RETURN
730 :
740 '-------------------------------------
750 R$="H":IF M%>6 THEN R$="CPC":GOTO 770
760 FOR J%=1 TO Q%:Y(J%)=T(J%):NEXT:GOTO 790
770 FOR J%=1 TO 6:Y(J%)=T(J%+6):Y(J%+6)=T(J%):NEXT
780 Y(P%)=T(Q%):Y(Q%)=T(P%):M%=M%-6
790 C%=M%:N%=Y(C%):FOR J%=1 TO N%:C%=C%+1
800 IF C%=P% THEN C%=1
810 IF C%=M% THEN C%=C%+1:GOTO 800
820 Y(C%)=Y(C%)+1:NEXT:Y(M%)=0:L%=C%
830 IF L%<7 OR Y(L%)>3 OR Y(L%)<2 THEN 850
840 Y(P%)=Y(P%)+Y(L%):Y(L%)=0:L%=L%-1:GOTO 830
850 S%=0:FOR J%=7 TO D%:S%=S%+Y(J%):NEXT
860 IF S%=0 THEN M%=-2:RETURN
870 IF R$="H" THEN FOR J%=1 TO Q%:T(J%)=Y(J%):NEXT:RETURN
880 FOR J%=1 TO 6:T(J%)=Y(J%+6):T(J%+6)=Y(J%):NEXT
890 T(Q%)=Y(P%):T(P%)=Y(Q%):RETURN
900 :
910 '========== JEU DU CPC 464 ========
920 :
930 FOR A%=1 TO 6:M%=A%+6:IF B(M%)=0 THEN U(A%)=-F%:GOTO 1100
940 FOR J%=1 TO Q%:T(J%)=B(J%):NEXT:GOSUB 710
950 IF M%<0 THEN U(A%)=-F%:GOTO 1100
960 IF T(Q%)>23 THEN M%=A%+6:RETURN
970 FOR J%=1 TO Q%:W(J%)=T(J%):NEXT:FOR K%=1 TO 6
980 IF T(K%)=0 THEN V(K%)=F%:GOTO 1080
990 FOR J%=1 TO Q%:T(J%)=W(J%):NEXT:M%=K%:GOSUB 710
1000 IF M%<0 THEN V(K%)=F%:GOTO 1080
```

```
1010 XA=0:XB=0.05:XC=0:XD=0:FOR J%=7 TO D%
1020 XB=XB+T(J%):IF T(J%)>0 THEN XA=XA+1
1030 IF T(J%)<3 THEN XC=XC+1
1040 IF T(J%)>XD THEN XD=T(J%)
1050 NEXT:XE=XB:FOR J%=1 TO 6:XE=XE+T(J%):NEXT
1060 XA=XA/6:XD=1-XD/XB:XC=1-XC/6:XB=XB/XE
1070 V(K%)=ZA*(XA+XB)+ZB*(XC+XD)+T(Q%)+B(P%)-B(Q%)-T(P%)
1080 NEXT:U(A%)=F%:FOR J%=1 TO 6:IF V(J%)<U(A%) THEN U(A%)=V(J%)
1090 NEXT
1100 PRINT".";:NEXT:M%=0:XA=-F%:FOR J%=1 TO 6
1110 IF(U(J%)>XA) THEN XA=U(J%):M%=J%+6
1120 NEXT:RETURN
1130 :
1140 '========== AFFICH. DES PIONS =====
1150 PEN 2:LOCATE 5,5:PRINT"CPC":PEN 3
1160 FOR J%=0 TO 5:LOCATE 4*J%+10,9:PRINT B(12-J%);
1170 IF B(12-J%)=0 THEN PEN 0:PRINT CHR$(8)+CHR$(8)+STRING$(2,143);:PEN 3

1180 IF B(12-J%)>9 THEN PEN 1:PRINT CHR$(8)+CHR$(143);:PEN 3
1190 NEXT:PEN 3:LOCATE 5,11:PRINT B(Q%)
1200 LOCATE 1,13:FOR J%=0 TO 5
1210 LOCATE 4*J%+10,13:PRINT B(J%+1);
1220 IF B(J%+1)=0 THEN PEN 0:PRINT CHR$(8)+CHR$(8)+STRING$(2,143);:PEN 3
1230 IF B(J%+1)>9 THEN PEN 1:PRINT CHR$(8)+CHR$(143);:PEN 3
1240 NEXT:PEN 3 :LOCATE 34,11:PRINT B(P%):PEN 2
1250 LOCATE 34,17:PRINT"VOUS"
1260 RETURN
1270 :
1280 '======== TRACE DU JEU ============
1290 C%=4:L%=7:GOSUB 1770
1300 PRINT CHR$(30);:PEN 2:FOR I%=1 TO 3:PRINT TAB(16)STRING$(11,143):NEX
T
1310 PRINT CHR$(30)+CHR$(22)+CHR$(1);:PEN 1:FOR I%=1 TO 3:PRINT TAB(16)ST
RING$(11,207):NEXT:LOCATE 17,2:PEN 3:PRINT"A W A R I":PRINT CHR$(22)+CHR$
(0)
1320 PEN 2:LOCATE 11,5:FOR J%=0 TO 5:PRINT TAB(4*J%+10);12-J%;:NEXT
1330 LOCATE 11,15:FOR J%=0 TO 5:PRINT TAB(4*J%+10);J%+1;:NEXT
1340 INK 1,3:INK 2,13:INK 3,25
1350 RETURN
1360 :
1370 '====== PAS DE MOUVEMENT ==========
1380 PEN 3:LOCATE 10,19: PRINT"PAS DE MOUVEMENT LEGAL."
```

```
1390 IF B(P%)=B(Q%) THEN PRINT:PRINT"          PARTIE NULLE.":GOTO 570
1400 IF B(Q%)>B(P%) THEN 380 ELSE 560
1410 REM ============ O/N ==============
1420 R$=UPPER$(INKEY$):IF R$<>"" THEN 1420
1430 R$=UPPER$(INKEY$):IF R$="" THEN 1430
1440 RETURN
1450 :
1460 '======== PRESENTATION ==========
1470 MODE 1:INK 0,0:INK 1,3:INK 2,13:INK 3,24:BORDER 0
1480 SYMBOL 251,&0,&0,&3C,&3C,&3C,&3C,&0,&0
1490 SYMBOL 252,&1F,&7F,&7F,&FF,&FF,&FF,&FF,&FF
1500 SYMBOL 253,&F8,&FE,&FE,&FF,&FF,&FF,&FF,&FF
1510 SYMBOL 254,&FF,&FF,&FF,&FF,&FF,&FE,&FE,&F8
1520 SYMBOL 255,&FF,&FF,&FF,&FF,&FF,&7F,&7F,&1F
1530 A$=CHR$(143)+STRING$(12,32)+CHR$(143):PEN 1:LOCATE 14,1:FOR I%=1 TO
2:PRINT TAB(14) A$;:NEXT
1540 MID$(A$,6,4)=STRING$(4,143):PRINT TAB(14) A$
1550 MID$(A$,5,6)=STRING$(6,143):PRINT TAB(14) A$
1560 PRINT TAB(14)STRING$(14,143)
1570 FOR I%=1 TO 6: PRINT TAB(17)STRING$(8,143):NEXT
1580 FOR I%=1 TO 3:PRINT TAB(18)STRING$(6,143):NEXT
1590 PRINT TAB(19)STRING$(4,143)
1600 PRINT TAB(20)STRING$(2,143)
1610 A$=CHR$(214)+CHR$(143)+CHR$(215):PRINT CHR$(22)+CHR$(1);:LOCATE 18,6
:PEN 3:PRINT A$+A$
1620 A$=CHR$(222)+CHR$(207)+CHR$(223):LOCATE 18,6:PEN 2:PRINT A$+A$
1630 PRINT TAB(18)STRING$(6,143)
1640 A$=CHR$(213)+CHR$(143)+CHR$(212):PRINT TAB(18)A$+A$
1650 LOCATE 19,7:PEN 0:PRINT CHR$(231):LOCATE 22,7:PRINT CHR$(231)
1660 LOCATE 19,12:PEN 3:PRINT STRING$(4,196):LOCATE 19,13:PRINT STRING$(4
,196)
1670 PRINT CHR$(22)+CHR$(0);
1680 PEN 1:LOCATE 16,20:PRINT CHR$(252)+STRING$(9,143)+CHR$(253)TAB(16)ST
RING$(11,143)TAB(16)CHR$(255)+STRING$(9,143)+CHR$(254)
1690 PEN 3:LOCATE 17,21:PRINT CHR$(22)+CHR$(1)"A W A R I"+CHR$(22)+CHR$(0
)
1700 PEN 2:LOCATE 2,24:PRINT"VOULEZ-VOUS LA REGLE DU JEU ? (O/N)"
1710 GOSUB 1410
1720 IF R$<>"O" THEN RETURN
1730 C%=4:L%=3:GOSUB 1770
1740 INK 1,3:INK 2,13:INK 3,25
1750 GOSUB 2020:RETURN
```

```
1760 :
1770 '====== TRACE DU TABLIER ==========
1780 CLS:INK 0,0:INK 1,0:INK 2,0:INK 3,0
1790 PEN 1:LOCATE C%,L%:PRINT CHR$(252)+STRING$(33,143)+CHR$(253);:FOR I%
=1 TO 7:PRINT TAB(4)STRING$(35,143)::NEXT:PRINT TAB(4)CHR$(255)+STRING$(3
3,143)+CHR$(254)
1800 PRINT CHR$(22)+CHR$(1)
1810  PEN 0:LOCATE C%,L%+1:PRINT CHR$(9)+CHR$(252)+STRING$(2,143)+CHR$(25
3);:FOR I%=1 TO 6:PRINT CHR$(9)+CHR$(252)+ CHR$(143)+ CHR$(253);:NEXT:PRI
NT CHR$(9)+CHR$(252)+STRING$(2,143)+CHR$(253)
1820 PRINT TAB(4)CHR$(9)+STRING$(4,143);:FOR I%=1 TO 6:PRINT CHR$(9)+STRI
NG$(3,143);:NEXT:PRINT CHR$(9)+STRING$(4,143)
1830 PRINT TAB(4)CHR$(9)+STRING$(4,143);:FOR I%=1 TO 6:PRINT CHR$(9)+CHR$
(255)+CHR$(143)+CHR$(254);:NEXT :PRINT CHR$(9)+STRING$(4,143);:PRINT TAB(
34)STRING$(4,143)
1840 PRINT TAB(4)CHR$(11)+CHR$(9)+STRING$(4,143)
1850 LOCATE C%,L%+5:PRINT CHR$(9)+STRING$(4,143);:FOR I%=1 TO 6:PRINT CHR
$(9)+CHR$(252)+CHR$(143)+ CHR$(253);:NEXT:PRINT CHR$(9)+ STRING$(4,143)
1860 PRINT TAB(4)CHR$(9)+STRING$(4,143);:FOR I%=1 TO 6:PRINT CHR$(9)+STRI
NG$(3,143);:NEXT:PRINT CHR$(9)+STRING$(4,143)
1870 PRINT TAB(4) CHR$(9)+CHR$(255)+STRING$(2,143)+CHR$(254);:FOR I%=1 TO
 6:PRINT CHR$(9)+CHR$(255)+CHR$(143)+CHR$(254);:NEXT:PRINT CHR$(9)+CHR$(2
55)+STRING$(2,143)+CHR$(254)
1880 PRINT CHR$(22)+CHR$(0)
1890 'INK 0,0:INK 1,3:INK 2,13:INK 3,25
1900 RETURN
1910 :
1920 '========== 4 GRAINES / CASE ======
1930 PRINT CHR$(30):PRINT CHR$(22)+CHR$(1);
1940 PEN 3:L%=8:FOR C%=10 TO 30 STEP 4:GOSUB 1980:FOR I%=1 TO 150:NEXT:NE
XT
1950 L%=4:FOR C%=30 TO 10 STEP -4:GOSUB 1980:FOR I%=1 TO 150:NEXT:NEXT
1960 PRINT CHR$(22)+CHR$(0):PEN 2
1970 RETURN
1980 LOCATE C%,L%:PRINT CHR$(136):LOCATE C%,L%+2:PRINT CHR$(130):LOCATE C
%+2,L%+2:PRINT CHR$(129):LOCATE C%+2,L%:PRINT CHR$(132):SOUND 7,50,2,12:F
OR I%=1 TO 50:NEXT:RETURN
1990 :
2000 '========= APPUI TOUCHE ===========
2010 LOCATE 33,25:PEN 3:PRINT"APPUYEZ";:CALL &BB18:LOCATE 33,25:PEN 0:PRI
NT STRING$(7,143);: PEN 2:RETURN
2020 '======= REGLES DU JEU ============
```

```
2030 PEN 2:LOCATE 1,16:PRINT"LE PLATEAU DE JEU COMPORTE 12 ALVEOLES"
2040 GOSUB 2000
2050 N%=1:FOR I%=10 TO 30 STEP 4:LOCATE I%,12:PRINT N%:SOUND 7,200,5,13:F
OR J=1 TO 150:NEXT:N%=N%+1:NEXT
2060 FOR I%=30 TO 10 STEP -4:LOCATE I%,2:PRINT N%:SOUND 7,200,5,13:FOR J=
1 TO 150:NEXT:N%=N%+1:NEXT
2070 LOCATE 1,18:PRINT"REMPLIES AU DEPART DE 4 GRAINES CHACUNE.":PRINT
2080 GOSUB 2000:GOSUB 1920
2090 LOCATE 1,20:PRINT"LES 6 ALVEOLES DU BAS ( NUMEROS 1 a 6 )":PRINT
2100 PRINT"CONSTITUENT VOTRE CAMP. CELLES DU HAUT":PRINT
2110 PRINT"SONT CELLES DE VOTRE ADVERSAIRE."
2120 GOSUB 2000:WINDOW #1,1,40,13,25:CLS#1
2130 LOCATE 1,14
2140 PRINT"CHAQUE JOUEUR A TOUR DE ROLE PREND LE":PRINT
2150 PRINT"CONTENU D'UNE CASE DE SON CAMP ET SEME":PRINT
2160 PRINT"TOUTES LES GRAINES QU'ELLE CONTIENT UNE":PRINT
2170 PRINT"PAR UNE DANS CHAQUE CASE EN TOURNANT":PRINT
2180 PRINT"DANS LE SENS INVERSE DES AIGUILLES":PRINT:PRINT"D'UNE MONTRE."
2190 GOSUB 2000:CLS #1
2200 LOCATE 1,16 :PRINT"EXEMPLE DE DISTRIBUTION:  CASE N"+CHR$(167)+"4."
2210 LOCATE 23,13:PEN 3:PRINT CHR$(240)
2220 FOR I=1 TO 500:NEXT
2230 PEN 1
2240 PRINT CHR$(22)+CHR$(1):LOCATE 22,8:PEN 0:PRINT CHR$(136)+" "+CHR$(13
2):LOCATE 22,10:PRINT CHR$(130)+" "+CHR$(129)
2250 FOR I=1 TO 800:NEXT
2260 LOCATE 27,9:PEN 3:PRINT CHR$(251):SOUND 7,100,10,12
2270 FOR I=1 TO 800:NEXT
2280 LOCATE 31,9:PEN 3:PRINT CHR$(251):SOUND 7,100,10,12
2290 FOR I=1 TO 800:NEXT
2300 LOCATE 31,5:PEN 3:PRINT CHR$(251):SOUND 7,100,10,12
2310 FOR I=1 TO 800:NEXT
2320 LOCATE 27,5:PEN 3:PRINT CHR$(251):SOUND 7,100,10,12
2330 PEN 2:LOCATE 1,20:PRINT"SI LA CASE CONTIENT PLUS DE 11 GRAINES":PRIN
T
2340 PRINT"ON EFFECTUE UN SECOND TOUR EN SAUTANT":PRINT
2350 PRINT"SA CASE DE DEPART."
2360 GOSUB 2000:CLS #1
2370 LOCATE 1,14:PEN 3:PRINT"PRISE:":PEN 2
2380 PRINT:PRINT"SI LA DERNIERE GRAINE DISTRIBUEE TOMBE"
2390 PRINT:PRINT"DANS UNE CASE ADVERSE CONTENANT DEJA "
2400 PRINT:PRINT"UNE OU DEUX GRAINES,LE JOUEUR PREND"
```

```
2410 PRINT:PRINT"TOUT LE CONTENU DE CETTE CASE (2 OU 3)"
2420 GOSUB 2000:CLS #1
2430 LOCATE 1,14:PRINT"SI LA CASE PRECEDANT CETTE CASE VIDEE"
2440 PRINT:PRINT"CONTIENT 2 OU 3 GRAINES,CES GRAINES SONT"
2450 PRINT"PRISES ELLES AUSSI ET AINSI DE SUITE"
2460 PRINT:PRINT"EN REMONTANT JUSQU'A CE QU'ON ARRIVE SUR"
2470 PRINT"UNE CASE NE CONTENANT NI 2 NI 3 GRAINES"
2480 PRINT:PRINT"OU DANS SON PROPRE CAMP."
2490 GOSUB 2000:CLS #1
2500 LOCATE 1,16:PEN 3:PRINT"DON DE GRAINES:"
2510 PRINT:PRINT:PEN 2:PRINT"SI L'ADVERSAIRE N'A PLUS DE GRAINES"
2520 PRINT:PRINT"IL FAUT LUI EN DONNER OBLIGATOIREMENT."
2530 PRINT:PRINT"SI C'EST IMPOSSIBLE,LA PARTIE S'ARRETE."
2540 GOSUB 2000:CLS #1
2550 LOCATE 1,18:PEN 3:PRINT"GAIN DE LA PARTIE":PEN 2
2560 PRINT:PRINT: PRINT:PRINT"LE PREMIER A AVOIR 24 GRAINES A GAGNE."
2570 GOSUB 2000:CLS #1
2580 LOCATE 1,18:PRINT"LE PROGRAMME PERMET LE RETOUR EN ARRIERE":PRINT"AV
EC OU NON LA REPRISE D'UN COUP AVEC":PRINT:PRINT"LES TOUCHES "+CHR$(242)+
" ET "+CHR$(243)
2590 GOSUB 2000:RETURN
2600 REM ===== RETOUR EN ARRIERE ======
2610 IF COUP%=1 THEN 400
2620 LOCATE 1,21:PRINT D$
2630 COUP%=COUP%-1
2640 GOSUB 2690
2650 INTER%=AR(COUP%,15)
2660 GOTO 400
2670 :
2680 '------- TRANSFERTS -------------
2690 FOR I%=1 TO Q%:B(I%)=AR(COUP%,I%):NEXT:GOSUB 1140:RETURN
2700 COUP%=COUP%+1:COUPMAX%=COUP%:IF COUP%=300 THEN IF B(Q%)>B(P%) GOTO 3
80 ELSE IF B(P%)>B(Q%) THEN GOTO 560 ELSE LOCATE 1,21:PEN 3:PRINT"
   MATCH NUL ....     ":GOTO 570
2710 FOR I%=1 TO Q%:AR(COUP%,I%)=B(I%):NEXT
2720 RETURN
2730 '========= avance ================
2740 IF COUP%=COUPMAX% THEN 400
2750 COUP%=COUP%+1
2770 INTER%=AR(COUP%,15)
2780 GOSUB 2690
2790 GOTO 400
```

3

CHAR D'ASSAUT

Emmanuel ALLINGRY

Vous conduisez un char d'assaut en pays ennemi. Votre mission est de détruire les fantassins qui viennent à votre rencontre, mais il faudra en permanence tenter d'éviter les mines. En outre, votre carburant est limité et, pour éviter de tomber en panne d'essence, vous devrez périodiquement vous ravitailler en attrapant des bidons disposés sur votre parcours. Bien plus grave encore, votre moteur s'use au fil des kilomètres parcourus et devient de plus en plus glouton.

```
10 SYMBOL AFTER 42
20 SYMBOL 42.0,16.84.56,254.56.84.16
30 SYMBOL 252,24,36,60,60,60,60.60,24
40 SYMBOL 171,16,16,84,124,124,124,124,84
50 GOTO 570
60 MODE 1
70 INK 0,15:INK 1,0:BORDER 15
80 PEN 1
90 PRINT    "           VOUS         "
100 PRINT   "          CONDUISEZ "
110 PRINT   "        UN CHAR D'ASSAUT"
120 PRINT   "         EN PAYS ENNEMI."
130 PRINT""
140 PRINT   "         VOTRE MISSION  "
150 PRINT   "         EST D'ECRASER"
160 PRINT   "      LES FANTASSINS ENNEMIS."
170 PRINT""
180  PRINT      "        MAIS ATTENTION "
190 PRINT   "       VOUS DEVEZ EVITER"
200 PRINT       "         LES MINES."
210 PRINT       " "
```

```
220 PRINT        "           EN OUTRE"
230   PRINT           "          VOTRE CARBURANT"
240 PRINT "          EST LIMITE."
250 PRINT""
260 PRINT "         VOUS DEVEZ DONC"
270 PRINT "        VOUS RAVITAILLER"
280 PRINT "          EN ATTRAPANT"
290 PRINT "           DES BIDONS"
300 PRINT"         SUR VOTRE PARCOURS."
310 PRINT""
320 PRINT"Pour continuer,appuyez sur C"
330 IF INKEY(62)=0 THEN 350
340 GOTO 330
350 MODE 1
360 PRINT"       AU COURS"
370 PRINT"    DES KILOMETRES"
380 PRINT"   VOTRE MOTEUR S'USE
390 PRINT"      ET CONSOMME
400 PRINT"      D'AVANTAGE."
410 PRINT"":PRINT"":PRINT""
420 PRINT"ATTEINDRE CERTAINS SCORES AJOUTE                   DU CARBURANT
"
430 PRINT"":PRINT"":PRINT"":PRINT"":PRINT""
440 PRINT"Pour continuer,appuyez sur C"
450 IF INKEY(62)=0 THEN MODE 1:GOTO 470
460 GOTO 450
470 LOCATE 10.2:  PRINT CHR$(171),"vous"
480 LOCATE 10,4 :   PRINT CHR$(248),"fantassin"
490 LOCATE 10.6: PRINT CHR$(42),"mine"
500 LOCATE 10,8: PRINT CHR$(252),"bidon"
510 PEN 3: LOCATE 10,10 : PRINT CHR$(248),"lieutenant":PEN 1
520 LOCATE 10,15:PRINT CHR$(242):LOCATE 13,15:PRINT"gauche"
530 LOCATE 10,17:PRINT CHR$(243):LOCATE 13,17: PRINT"droite"
540 LOCATE 1,20:PRINT"Pour continuer,appuyez sur C"
550 IF INKEY(62)=0 THEN 570
560 GOTO 550
570 INK 0,15:INK 1,0,26:BORDER 15:MODE 1
580 PEN 1
590 LOCATE 12 ,14: PRINT"CHARS D'ASSAUT"
600 PEN 3
610 PLOT 175,170:DRAW 399,170
```

```
620 LOCATE 20,25:PRINT"Pour jouer,appuyez sur J"
630 LOCATE 20,20:PRINT"Pour des precisions,appuyez sur P"
640 IF INKEY(45)=0 THEN  670
650 IF INKEY(27)=0 THEN  60
660 GOTO 640
670 MODE 0:INK 1,0:INK 0,15:BORDER 15:INK 4,1
680 l=10:k=((20-l)/2)+0.5
690 car=200
700 m=k:n=l+k
710 v=RND(1)*l+k:w=RND(1)*l+k:x=RND(1)*l+k:y=RND(1)*l+k:z=RND(1)*l+k:bid=
RND(1)*l+k:mie=RND(1)*l+k:mif=RND(1)*l+k
720 PEN 0
730 f=l/2+k-0.5: LOCATE f,20:PRINT CHR$(171)
740 PLOT 90,25,4:DRAW 130,375,4:PLOT 555,25,4:DRAW 515,375,4
750 a=8:b=11:c=14:d=17:e=20:bi=16:mi=13:mi2=10:h=1.5:g=0:sc=0
760 PEN 4
770 LOCATE 11,2:PRINT"carburant"
780 REM
790 PEN 3: LOCATE v,a:PRINT CHR$(248)
800 PEN 4
810 LOCATE v,a-1:PRINT CHR$(32)
820 a=a+1
830  IF a=21 THEN a=8:LOCATE v,20:PRINT CHR$(32):IF f=ROUND(v) THEN SOUND
 1,200,20:v=RND(1)*l+k:sc=sc+50:LOCATE 2,2:PRINT"score",sc ELSE v=RND(1)*
l+k
840 GOSUB 1340
850 PEN 5
860 LOCATE MIf,mi2:PRINT CHR$(42)
870 PEN 4
880 LOCATE MIf,mi2-1:PRINT CHR$(32)
890 mi2=mi2+1
900 IF mi2=21 THEN mi2=8:LOCATE MIF,20:PRINT CHR$(32):IF f=ROUND(MIF) THE
N SOUND 1,1200,20 :GOTO 1390:mif=RND(1)*l+k ELSE mif=RND(1)*l+k
910 PEN 7
920 LOCATE w,b:PRINT CHR$(248)
930 PEN 4
940 LOCATE w,b-1:PRINT CHR$(32)
950 b=b+1
960 IF b=21 THEN b=8:LOCATE w,20:PRINT CHR$(32):IF f=ROUND(w) THEN SOUND
1,200,20 :w=RND(1)*l+k:sc=sc+15:LOCATE 2,2:PRINT"score",sc  ELSE w=RND(1)
*l+k
970 PEN 5
```

```
980 LOCATE MIe,mi:PRINT CHR$(42)
990 PEN 4
1000 LOCATE MIe,mi-1:PRINT CHR$(32)
1010 mi=mi+1
1020 IF mi=21 THEN mi=8 :LOCATE MIe,20:PRINT CHR$(32):IF f=ROUND(MIe) THE
N mie=RND(1)*l+k :SOUND 1,1200,20 :GOTO 1390:ELSE mie=RND(1)*l+k
1030 PEN 7
1040 LOCATE x,c:PRINT CHR$(248)
1050 PEN 4
1060 LOCATE x,c-1:PRINT CHR$(32)
1070 GOSUB 1340
1080 c=c+1
1090 IF c=21 THEN c=8:LOCATE x,20:PRINT CHR$(32):IF ROUND(x)=f THEN SOUND
 1,200,20:x=RND(1)*l+k:sc=sc+15:LOCATE 2,2:PRINT"score",sc  ELSE x=RND(1)
*l+k
1100 PEN 7
1110 LOCATE v,d:PRINT CHR$(248)
1120 PEN 4
1130 LOCATE v,d-1:PRINT CHR$(32)
1140 d=d+1
1150 IF d=21 THEN d=8:LOCATE v,20:PRINT CHR$(32):IF ROUND(v)=f THEN SOUND
 1,200,20:v=RND(1)*l+k:sc=sc+15:LOCATE 2,2:PRINT"score",sc  ELSE v=RND(1)
*l+k
1160 PEN 7
1170 LOCATE z,e:PRINT CHR$(248)
1180 PEN 4
1190 LOCATE z,e-1:PRINT CHR$(32)
1200 e=e+1
1210 IF e=21 THEN e=8:LOCATE z,20:PRINT CHR$(32):IF ROUND(z)=f THEN SOUND
 1,200,20:z=RND(1)*l+k :sc=sc+15:LOCATE 2,2:PRINT"score",sc ELSE z=RND(1)
*l+k
1220 GOSUB 1340
1230 PEN 10
1240 LOCATE bid,bi:PRINT CHR$(252)
1250 PEN 4
1260 LOCATE bid,bi-1:PRINT CHR$(32)
1270 bi=bi+1
1280 IF bi=21 THEN bi=8:LOCATE bid,20:PRINT CHR$(32):IF ROUND(bid)=f THEN
 SOUND 1,500,20:bid=RND(1)*l+k :car=car+35:LOCATE 10,3:PRINT INT(car) ELS
E bid=RND(1)*l+k
1290 g=g+1:IF g/10=INT(g/10) THEN h=h+0.1
1300 car=car-h:IF car<0 THEN 1460 ELSE LOCATE 10,3:PRINT  INT(car)
```

```
1310 IF sc>1000 AND l1=0 THEN l1=1:car=car+500
1320 IF sc>2500 AND l2=0 THEN l2=1:car=car+1000
1330 GOTO 780
1340 IF INKEY(8)=0 THEN f=f-1:IF f<m-1 THEN 1530
1350 LOCATE f,20:PRINT CHR$(171):LOCATE f+1,20:PRINT CHR$(32)
1360 IF INKEY(1)=0 THEN f=f+1:IF f>n+1 THEN 1530
1370 LOCATE f,20:PRINT CHR$(171):LOCATE f-1,20:PRINT CHR$(32)
1380 RETURN
1390 MODE 1: PEN 3:LOCATE 15,20:PRINT"PERDU"
1400 IF sc>sc1 THEN sc1=sc
1410 PRINT"":PRINT"MEILLEUR SCORE",sc1
1420 PEN 1
1430 LOCATE 1,25:PRINT"VOUS AVEZ RENCONTRE UNE MINE"
1440 LOCATE 1,1:PRINT"score",sc
1450 FOR g=0 TO 5000:NEXT:GOTO 570
1460  MODE 1: PEN 3:LOCATE 15,20:PRINT"PERDU"
1470 IF sc>sc1 THEN sc1=sc
1480 PRINT"":PRINT"MEILLEUR SCORE",sc1
1490 PEN 1
1500 LOCATE 1,1:PRINT"score",sc
1510 LOCATE 1,25:PRINT"VOUS ETES TOMBE EN PANNE DE CARBURANT"
1520 FOR g=0 TO 5000:NEXT:GOTO 570
1530 MODE 1: PEN 3:LOCATE 15,20:PRINT"PERDU"
1540 IF sc>sc1 THEN sc1=sc
1550 PRINT"":PRINT"MEILLEUR SCORE",sc1
1560 PEN 1
1570 LOCATE 1,24:PRINT"VOUS ETES SORTI DE VOTRE ROUTE ET VOUS  VOUS ETES
ENLISE"
1580 LOCATE 1,1:PRINT"score",sc
1590 FOR g=0 TO 5000:NEXT:GOTO 570
```

3

CONSOMMATION DE CARBURANT

Michel ARCHAMBAULT

Pour calculer une consommation en litres aux 100 km, une simple calculatrice ou dix lignes de Basic sont suffisants. Alors, pourquoi un programme aussi long ? C'est le super luxe !
La saisie est facile car elle supprime le côté fastidieux et les risques d'étourderies de la chose ; enfin, le grand PLUS est l'enregistrement automatique, sur cassette ou disquette, des données.

Ceci permet d'avoir à l'écran **l'évolution** de la consommation au cours des mois et des années (très instructif !). C'est donc un logiciel de comptabilité automatique très spéciale qui permet de "surveiller" plusieurs véhicules.
Il a été conçu pour qu'il soit adaptable à la fois sur cassette ou disquette sans la moindre modification.

PETIT DESCRIPTIF

Chaque fois que vous prenez du carburant, vous notez sur un carnet de bord les litres, le kilométrage au compteur, la date et la lettre P s'il s'agissait d'un plein.
A des intervalles quelconques, par exemple tous les mois, vous entrez certaines de ces données au clavier de votre AMSTRAD. C'est très rapide. L'écran vous affiche alors les éléments suivants :
— Entre chaque plein, la consommation en litres, en litres aux 100 km et la distance en kilomètres.
— La même chose mais cumuls et moyennes depuis le début de la saisie en cours.
— Tableau final : résultats de la saisie + ceux de la saisie précédente + moyennes et cumuls depuis la date des **débuts** des relevés.
Ces données sont alors enregistrées sur un fichier portant le nom du véhicule en question.

A la saisie suivante (un ou deux mois plus tard), on entre seulement le nom de la voiture ; le programme charge alors le fichier correspondant et vous invite à la suite de la saisie, là où vous l'aviez arrêtée. Pas d'erreurs possibles.
De nombreuses lignes BASIC sont consacrées à la clarté des tableaux et à guetter les étourderies d'un opérateur distrait. D'accord, c'est plus long à taper, mais après, on est tranquille.
C'est un logiciel utilitaire, pas un programme gadget. Il faut choisir...

LE MODE D'EMPLOI

Après un titre quelque peu évocateur, on doit entrer le nom de la voiture : huit caractères maxi, sans espace ni signe de ponctuation et commençant par une lettre. Si vous avez une Peugeot 205, n'entrez pas "205", mais plutôt "P205". Apparaît alors le menu principal, carrefour du programme : chargement (lecture) du fichier, saisie, sauvegarde et "première fois", autrement dit, l'ouverture d'un fichier voiture. On y demande le kilométrage et la date du **premier** plein ; les litres versés, on s'en moque. Cette opération enchaîne sur la saisie.
Les fois suivantes, on démarre directement sur l'option C (chargement du fichier). C'est rapide puisqu'il ne contient que neuf valeurs. L'écran vous affiche alors la date et le kilométrage du dernier plein enregistré, afin de vous situer dans votre carnet. Puis vous êtes invité à entrer la date et le kilométrage de la ligne suivante du carnet, même s'il ne s'agit pas d'un plein. Puis commence la saisie.

LE TABLEAU DE SAISIE

Il est constitué de six WINDOWS : un pour les entrées, deux pour les résultats, et trois pour les légendes de colonnes, afin de les protéger du scrolling.
La fenêtre des entrées est celle du haut ; trois colonnes où le curseur ira se positionner automatiquement.
a) Entrée des litres. Enter.
b) Le curseur va sur la colonne "Plein". Si ce n'est pas le cas, faites Enter, et le curseur revient sur l'entrée des litres. Si c'est un plein, tapez "P". Le curseur va alors dans la colonne "KM" où vous entrez le kilométrage à ce plein ; et Enter.
c) Aussitôt les résultats apparaissent dans les fenêtres inférieures : à gauche, concernant cette "étape" de plein à plein. A droite concernant cumuls et consommation moyenne depuis le début de la saisie.
d) Le curseur est revenu sur l'entrée litres, on poursuit de même.
e) Vous venez de constater une erreur sur les litres entrés, vous avez déjà fait Enter, et le curseur est dans la colonne "Plein" : tapez "E" et Enter ; la dernière entrée de litres est oubliée.

f) Fin de saisie. Le curseur attend une entrée de litres ; tapez "F" et Enter. Apparaît alors le...

TABLEAU DES BILANS

En haut de l'écran, le bilan de cette saisie (c'est donc la dernière ligne de la fenêtre de droite de l'écran de saisie). Au milieu, le bilan de la saisie précédente (avec ses dates). En bas, le bilan global calculé à partir de la toute première saisie, donc pouvant rassembler plusieurs années.

NOTE : Lors de cette première saisie (option "première fois"), il est normal que la "saisie précédente" n'apparaisse pas...
Ceci vu, on presse une touche quelconque et le nouveau fichier est enregistré. Retour au menu, Fin.

L'ENREGISTREMENT DES FICHIERS

Un nouveau fichier écrase l'ancien. Si vous avez un lecteur de disquettes, pas de problèmes ; sur une même face, il y a le programme CONSOM et autant de noms de fichiers qu'il y a de véhicules à "surveiller".

Avec magnétophone, une cassette pour le programme et autant de cassettes fichiers qu'il y a de véhicules (en débuts de cassettes). Toujours rebobiner la cassette fichier après lecture.

LE PROGRAMME

Il est en "structuré", les modules étant appelés à partir du menu (ligne 1000). En lignes 65000, nous avons rassemblé la liste alphabétique des variables (on est ainsi sûr de ne pas l'égarer...).
Les messages à l'écran sont avec minuscules accentuées, ce qui explique des signes bizarres dans le listing ; les accollades sont obtenues par SHIFT + crochets, elles donneront "é" et "è" ; le a commercial se transformera en "à".
CALL &BB06 bloque le programme jusqu'à ce que l'on presse une touche.
Le nom de la voiture, VOIT$, peut être entré indifféremment en majuscules ou en minuscules (ligne 760). A ce propos, nous avons débusqué **un bug de l'AMSTRAD** :
Il y avait une ligne 770 VOIT$= UPPER$(VOIT$) qui nous a joué la farce suivante : à la lecture du fichier, nous obtenions le message "fichier inconnu" (sur lecteur de disquettes), et pour cause ! Le nom "ASCONA" était devenu "OOOOAN" sur l'enregistrement... En supprimant cet UPPER$, tout est devenu normal. C'est inexpliqué, mais il faudra s'en souvenir.

```
10 'CONSOMMATION DE CARBURANT
20 'Michel ARCHAMBAULT--1985 / AMSTRAD
40 DEFINT I.J,M,N
50 GOSUB 51000:'AZERTY
60 INK 0,1:INK 1,24
70 PAPER 2:PEN 3:BORDER 15
100 'TITRE
110 MODE 0:CLS
120 LOCATE 5,4:PRINT "C O N S O M"
130 FOR I=0 TO 200 STEP 2:PLOT 0,I,9
140 DRAWR 640,0:NEXT:PEN 0
150 FOR I=1 TO 20:LOCATE I,7:PRINT "$":FOR J=1 TO 200:NEXT:NEXT
160 FOR I=1 TO 1000:NEXT
170 FOR Z=200 TO 15 STEP -0.5:PLOT 0,Z,2
180 DRAWR 640,0:NEXT:LOCATE 2,23:PRINT "M.Archambault-1985"
190 FOR I=1 TO 2000:NEXT
700 'VOITURE
710 MODE 1:CLS:PEN 3
720 LOCATE 12,5:PRINT "NOM DE LA VOITURE:":PEN 0
730 LOCATE 7,7:PRINT "(sans espace ni ponctuation)"
740 LOCATE 8,9:PRINT "(commencez par une lettre)"
750 LOCATE 17,12:PRINT STRING$(8,"."):LOCATE 17.12:INPUT "".VOIT$
760 IF VAL(VOIT$)>0 OR LEN(VOIT$)>8 THEN PRINT CHR$(7):GOTO 750
1000 ' MENU
1010 PEN 3:PAPER 2:CLS:FPF=0
1020 LOCATE 17,3:PRINT "M E N U":PEN 0
1030 LOCATE 12,6:PRINT "C : CHARGEMENT FICHIER"
1040 LOCATE 12,8:PRINT "E : ENTREE DES LITRES."
1050 LOCATE 12,10:PRINT "S : SAUVEGARDE FICHIER."
1060 LOCATE 12,12:PRINT "P : PREMIERE FOIS."
1070 LOCATE 12.14:PRINT "F : FIN."
1080 TEX$="CESPF":PEN 3:GOSUB 50000:PEN 0
1090 ON K GOSUB 2000,5000,10000,15000,20000
1100 GOTO 1000
2000 'CHARGEMENT DU FICHIER
2010 PAPER 0:PEN 3:BORDER 2:CLS
2020 LOCATE 6,6:PRINT "CHARGEMENT DU FICHIER VOITURE.":PEN 1
2030 LOCATE 19,12:PRINT "OK ?":TEX$="ON":GOSUB 50000:LOCATE 1,24:PRINT CH
R$(18)
2040 IF K=2 THEN 2900
2050 LOCATE 4,15:PRINT "Pr{parez la Cassette ou Disquette."
```

```
2060 LOCATE 9,17:PRINT "puis pressez une Touche...":CALL &BB06
2100 OPENIN VOIT$
2110 INPUT #9,DATO$,DATD$,DATF$,KO,KD,DSP,KF,LS,LTOT
2120 CLOSEIN
2200 CLS
2210 LOCATE 12,4:PEN 3:PRINT "VOITURE ";VOIT$;" :":PEN 2
2220 LOCATE 12,7:PRINT "Pr{c}dente Saisie:"
2230 LOCATE 10,10:PRINT "du ";DATD$;" au ";DATF$"
2240 LOCATE 10,12:PRINT "de";KD;"km @";KF;"km"
2250 LOCATE 19,16:PEN 1:PRINT "OK ?":TEX$="ON":GOSUB 50000
2260 IF K=2 THEN 2900
2900 RETURN
5000 'ENTREE DES DONNEES
5010 PEN 3:CLS:IF FPF=1 THEN 5200
5050 LOCATE 12,3:PRINT "ENTREE DES DONNEES:":PEN 0
5060 LOCATE 4,6:PRINT "Dernier Relev{:";DATF$;KF;"km"
5070 LOCATE 15,8:PRINT "Date D{part:":GOSUB 12000
5080 LOCATE 11,17:INPUT "km d{part: ",KD$:KD=VAL(KD$):IF KD<KF THEN 5080
5090 LOCATE 10,20:PRINT DAT$;" &";KD;"km : OK ?":TEX$="ON":GOSUB 50000
5100 IF K=2 THEN 5000
5110 DAT1$=DAT$:KMP=KF
5200 'TABLEAU DE SAISIE
5210 WINDOW #1,1,40,1,3:PAPER #1,1:PEN #1,0:CLS #1
5220 PRINT #1:PRINT #1,TAB(4);"Litres:";TAB(17);"PLEIN";TAB(28);"COMPTEUR
:"
5230 PRINT #1,TAB(4);"(FIN=F)";TAB(18);"(P)";TAB(30);"(km)"
5240 WINDOW #2,1,40,4,6:PAPER #2,1:PEN #2,0:CLS #2
5250 WINDOW #3,1,19,7,11:PAPER #3,2:PEN #3,0:CLS #3:PRINT #3
5260 PRINT #3,TAB(3);"Plein a Plein:":PRINT #3
5270 PRINT #3,TAB(3);"L";TAB(10);"km";TAB(15);"L/100"
5280 WINDOW #4,20,40,7,11:PAPER #4,0:PEN #4,3:CLS #4:PRINT #4
5290 PRINT #4,TAB(5);"Bilan cumul{:":PRINT #4
5300 PRINT #4,TAB(4);"L";TAB(10);"km";TAB(16);"L/100"
5310 WINDOW #5,1,19,11,25:PAPER #5,2:PEN #5,0:CLS #5
5320 WINDOW #6,20,40,11,25:PAPER #6,0:PEN #6,3:CLS #6
5330 BORDER 15
6000 'SAISIE
6005 N=1:M=1:LT=0:TOLT=0
6010 LOCATE #2,6,N:INPUT #2,"",L$:IF UPPER$(L$)="F" THEN 7000
6030 L=VAL(L$):IF L=0 THEN PRINT CHR$(7);:GOTO 6010
6040 LT=LT+L
6050 LOCATE #2,19,2:INPUT #2,"",P$:IF P$="" THEN N=N+1:GOTO 6010
```

```
6055 IF UPPER$(P$)="E" THEN LT=LT-L:N=N-1:GOTO 6010
6060 IF UPPER$(P$)<>"P" THEN 6050
6070 LOCATE #2,29,2:INPUT #2,"",KM$:KM=VAL(KM$)
6080 IF KM<=KMP THEN PRINT CHR$(7);:GOTO 6070
6100 DKM=KM-KMP:KMP=KM:CONP=100*LT/DKM
6110 TOLT=TOLT+LT:TKM=TKM+DKM:CONG=100*TOLT/TKM
6150 LOCATE #5,2,M:PRINT #5,USING"##.#";LT
6160 LOCATE #5,8,M:PRINT #5,USING"####";DKM
6170 LOCATE #5,15,M:PRINT #5,USING"##.#";CONP
6180 LOCATE #6,2,M:PRINT #6,USING"###.#";TOLT
6190 LOCATE #6,9,M:PRINT #6,USING"####";TKM
6200 LOCATE #6,17,M:PRINT #6,USING"##.#";CONG
6210 M=M+1:N=N+1:LT=0:GOTO 6010
7000 'TABLEAU FINAL
7010 PAPER 0:CLS
7020 LOCATE 8,6:PEN 1:PRINT "Date de ce dernier Plein :":PEN 2:GOSUB 1200
0
7030 CLS:BORDER 12:LOCATE 2,3:PRINT "A ce Relev{ du ";DAT1$;" au ";DAT$
7040 LOCATE 1,5:PRINT TKM;"km avec";ROUND(TOLT,1);"L.soit";ROUND(CONG,1);
"L/100 km"
7050 PRINT STRING$(40,"_"):IF FPF=1 THEN 7090
7060 PEN 2:LOCATE 1,8:PRINT "Relev{ pr{c}dent,du ";DATD$;" au ";DATF$
7070 LOCATE 1,10:PRINT DSP;"km avec";ROUND(LS,1);"L,soit";ROUND(100*LS/DS
P,1);"L/100 km"
7080 PRINT STRING$(40,"_")
7090 LS=TOLT:LTOT=LTOT+TOLT:DATD$=DAT1$:DATF$=DAT$:KF=KM:DSP=TKM
7100 PEN 3:LOCATE 3,14:PRINT "BILAN GENERAL du ";DATO$;" au ";DATF$
7110 LOCATE 1,16:PRINT KF-KO;"km avec";ROUND(LTOT,1);"L,soit";ROUND(100*L
TOT/(KF-KO),1);"L/100 km"
7120 PEN 1:LOCATE 8,20:PRINT "SAUVEGARDE DE CES VALEURS:"
7130 LOCATE 3,22:PRINT "Pr{parez la Cassette ou Disquette,"
7140 LOCATE 8,24:PRINT "puis pressez une Touche..."
7150 CALL &BB06
7160 GOSUB 10000
7170 RETURN
10000 ' ENREGISTREMENT DU FICHIER
10010 OPENOUT VOIT$
10020 WRITE #9,DATO$,DATD$,DATF$,KO.KD,DSP,KF.LS,LTOT
10030 CLOSEOUT
10900 RETURN
12000 'ENTREE DATE
12010 LOCATE 17,10:INPUT "jour: ",J$:J=VAL(J$):IF J<1 OR J>31 THEN 12010
```

```
12020 LOCATE 17,12:INPUT "mois: ",M$:M=VAL(M$):IF M<1 OR M>12 THEN 12020
12030 LOCATE 16,14:INPUT "ann{e: ",AN$:AN$=RIGHT$(AN$,2):IF VAL(AN$)<80 T
HEN 12030
12040 DAT$=J$+"/"+M$+"/"+AN$
12900 RETURN
15000 'PREMIERE FOIS
15010 BORDER 2:PAPER 1:PEN 0:CLS
15020 LOCATE 2,4:PRINT "Vous ouvrez un NOUVEAU fichier voiture."
15030 LOCATE 18,6:PRINT "OK ?":TEX$="ON":GOSUB 50000:IF K=2 THEN 15900
15035 CLS:LOCATE 9,2:PRINT"Au tout premier Plein:"
15040 LOCATE 8,4:INPUT"Kilom{trage D{part:",KO$
15050 KO=VAL(KO$):IF KO=0 THEN 15040
15060 LOCATE 16,7:PRINT "en Date du:":GOSUB 12000
15065 LOCATE 8,18:PRINT "N'ENTREZ PAS CES LITRES !":FOR I=1 TO 4000:NEXT
15070 DATO$=DAT$:DATD$=DATO$:DATF$=DATO$:DAT1$=DATO$
15080 KF=KO:LS=0:LTOT=0:KMP=KO:KD=KO
15090 FPF=1
15100 PAPER 0:GOSUB 5000
15900 RETURN
20000 'FINAL
20010 PAPER 0:PEN 1:CLS
20020 LOCATE 10,12:PRINT "VOUS POUVEZ ETEINDRE"
20030 LOCATE 11,14:PRINT "ou relancer par RUN.":PRINT:PRINT
20100 END
49999 END
50000 'REPONSE A UN MENU
50010 LT=LEN(TEX$):R$=""
50020 LOCATE 15-LT,24:PRINT "R{ponse (";
50030 FOR I=1 TO LT-1
50040 PRINT MID$(TEX$,I,1);",";:NEXT
50050 PRINT RIGHT$(TEX$,1);")";CHR$(154);CHR$(243);CHR$(207)
50060 TEX$=UPPER$(TEX$)
50070 WHILE R$="":R$=INKEY$:WEND
50080 R$=UPPER$(R$):K=INSTR(TEX$,R$)
50090 IF K=0 THEN R$="":PRINT CHR$(7);:GOTO 50070
50100 RETURN
51000 ' AZERTY ACCENTUE
51010 SYMBOL AFTER 6
51020 SYMBOL 64,96,48,120,12,124,204,118,0
51030 SYMBOL 91,0,56,108,56,0,0,0,0
51040 SYMBOL 92,0,0,60,102,96,102,60,24
51050 SYMBOL 93,60,96,60,102,60,6,60,0
```

```
51060 SYMBOL 123,12,24,60,102,126,96,60,0
51080 SYMBOL 125,48,24,60,102,126,96,60,0
51100 RETURN
65000 '------LEGENDE DES VARIABLES----
65001  CONG : CONSOMMATION GLOBALE
65002  CONP : CONSOM. DE PLEIN A PLEIN
65003  DAT$ : DATE AU CLAVIER
65004 DATO$ : DATE OUVERTURE FICHIER
65005 DAT1$ : DATE DEPART EN COURS
65006 DATD$ : DATE DEPART PRECEDENT
65007 DATF$ : DATE FIN DE SAISIE
65008  DKM  : DISTANCE ENTRE DEUX PLEINS
65009  DSP  : DISTANCE SAISIE PRECEDENTE
65010  FPF  : FLAG PREMIERE FOIS
65011  I    : UNITE DE COMPTAGE
65012  K    : INDICE DE MENU
65013  KO   : KM A L'OUVERTURE DU FICHIER
65014  KD   : KM DEPART SAISIE
65015  KF   : KM FIN DE SAISIE
65016  KM   : KM AU PLEIN
65017  KMP  : KM AU PLEIN PRECEDENT
65018  L    : LITRES ENTRES
65019  LS   : LITRES DE LA SAISIE
65020  LT   : LITRES ENTRE DEUX PLEINS
65021  LTOT : LITRES TOTAUX
65022  M,N  : COMPTAGES DE LIGNES
65023  TEX$ : SIGLES DU MENU
65024  TKM  : DISTANCE TOTALE EN SAISIE
65025  TOTL : LITRES SAISIE PRECEDENTE
65026 VOIT$ : NOM DE LA VOITURE
65027  Z    : UNITE DE COMPTAGE
65028 --------FIN DE LISTING--------
```

3

LE LECTEUR DE DISQUETTES POURQUOI ?

Michel ARCHAMBAULT

Beaucoup d'entre vous "lorgnent" depuis quelque temps vers le lecteur de disquettes (on dit aussi "DRIVE") AMSTRAD DDI-1, et ce en se disant "qu'aurais-je de plus ?". En fait, une foule de questions auxquelles nous allons répondre en essayant d'être clair, donc en évitant d'entrer dans des détails d'utilisation (d'autant plus qu'ils sont dans la notice du DDI-1...).

QU'Y A-T-IL DANS LA BOÎTE ?

Tout d'abord une **excellente notice en français** de six millimètres d'épaisseur, illustrée et facile à comprendre.
Un boîtier plat (l'interface) qui s'enfiche à l'arrière du CPC 464, et duquel sort un large câble plat. Le lecteur proprement dit, de la taille d'une cartouche de cigarettes, assez lourd, équipé d'un cordon secteur et avec, à l'arrière, un interrupteur marche-arrêt et un connecteur recevant le large câble plat. Une disquette enregistrée, numérotée, super précieuse (difficilement remplaçable), qui possède, sur une face, toutes les routines d'exploitation et sur l'autre, le langage LOGO (LOGO ? BOF...).
Il n'a a pas de disquette vierge. Pensez donc à ne pas revenir sans. Au moins deux : une qui sera un duplicata de la disquette précieuse (à ranger en lieu sûr) et une autre pour sauver vos programmes sur cassettes.

C'EST QUOI UNE DISQUETTE ?

C'est un boîtier plat et rigide mesurant 100 × 80 × 5 mm. C'est le "standard" trois pouces (à ne pas confondre avec le trois pouces et demi...). A l'intérieur, un disque mince recouvert recto-verso d'un vernis magnétique ; le même que sur les bandes magnétiques.

La suite rappelle un peu les avantages du disque 33 tours sur la cassette enregistrée : quand vous voulez écouter telle chanson, vous lisez sur l'étiquette que c'est l'air numéro cinq, et directement vous posez le bras de lecture sur cette plage. C'est l'ACCES DIRECT. Avec la version cassette, il vous faut d'abord passer (ou bobiner) les quatre précédents ! C'est l'ACCES SEQUENTIEL.
Revenons à notre disquette. Sur la face A, il y a, supposons, 28 programmes ; vous voulez utiliser le programme "BIDULE". Facile ! Vous tapez au clavier RUN "BIDULE". Que se passe-t-il alors ?
Le bras de lecture, supportant une tête magnétique, va d'abord aller lire la piste "catalogue"; c'est l'équivalent de l'étiquette centrale d'un 33 tours ; il y trouve le nom du programme avec son emplacement sur la face, par exemple "départ sur la piste n° 17, secteur n° 4". Le bras se place alors sur cette piste, et lorsque le secteur n° 4 passe devant la tête, il commence à charger le programme. Et à quelle allure !!! Un programme de 19 blocks (≃38 kilo-octets) se charge en neuf secondes et huit dixièmes !... à faire rêver.
Pourquoi est-ce beaucoup plus rapide que sur magnétophone ? Parce que ce dernier a été conçu au départ pour la musique, c'est-à-dire pour reproduire fidèlement des variations d'amplitude (de volume). Pas un lecteur de disquettes ; il ne connaît que le binaire, l'amplitude est 1 ou 0, c'est tout. On peut alors pousser la fréquence beaucoup plus haut.
Pour fixer les idées, disons que la disquette 3 pouces pour AMSTRAD comporte 40 pistes **concentriques,** numérotées de 0 à 39 ; chacune étant divisée en neuf secteurs de 512 octets chacun. Faisons le calcul :

$$512 \times 9 \times 40 = 184\ 320 \text{ octets (180 kilo-octets).}$$

Et comme les deux faces sont utilisables, cela nous fait 360 k-octets par disquette. Confortable, non ? Une disquette vierge vaut actuellement entre 58 et 75 francs selon les revendeurs. Plus chère que la cassette ? Non, car une disquette peut stocker beaucoup plus qu'une cassette. Disons que, globalement, c'est légèrement en faveur de la disquette face à la C60 à 12 francs (que l'on ne remplit jamais...).

C'EST QUOI, LE FORMATAGE ?

Vous venez d'acheter une disquette vierge ; elle l'est vraiment ! Car les pistes et les secteurs n'y sont pas. Donc, premier soin, il faut la "formater" : le "DRIVE" va tracer ses pistes de guidage de tête et ses repères de secteurs (magnétiquement bien sûr). Cette opération représente environ une minute par face. Ce traitement est définitif. La disquette est alors prête à recevoir vos enregistrements.

LA DISQUETTE, EST-ELLE FRAGILE ?

Beaucoup moins qu'une cassette ! D'abord, la disquette trois pouces est à l'abri des poussières et des traces de doigts, car la fente de lecture comporte un volet qui se referme lorsque l'on sort la disquette du DRIVE.
D'autre part, il n'y a aucune mécanique, donc plus de risque de coincements et de bourrages destructeurs comme avec les cassettes.
NOTE : Les disquettes 5 pouces 1/4 sont bien plus économiques ($\simeq$ 15 francs), mais sont un peu plus délicates à manipuler, et la mécanique de leurs DRIVES est plus chère ; d'où cette nouvelle orientation vers le trois pouces et demi et le trois pouces, notre standard, hélas, tout neuf encore.

ET LE REPERTOIRE PAR FACE ?

Tapez CAT et Enter : le DRIVE lit le catalogue (on dit aussi DIRECTORY) et l'affiche à l'écran. Nous avons le nom de chaque programme ou fichier, son type (Basic, binaire, etc.) et sa taille en kilo-octets. Le tout classé par ordre alphabétique. En bas de liste apparaît le nombre de kilo-octets encore disponibles.
Vous pouvez alors la recopier au crayon sur l'étiquette carton fournie avec l'emballage rigide transparent de la disquette.

A PART LA VITESSE, C'EST TOUT ?

Oh que non ! La vitesse d'enregistrement et de lecture, c'est certes le plus spectaculaire, mais il faut adjoindre trois autres aspects :
— le côté exploitation qui devient beaucoup plus souple, plus rationnel, plus sûr. En un mot, plus pratique (indépendamment de la vitesse) ;
— les débouchés de "l'accès direct", c'est-à-dire certains programmes utilitaires qui sont irréalisables en "accès séquentiel" (cassette) ;
— un apprentissage pratique à la micro-informatique professionnelle.
En effet, l'AMSTRAD a choisi le "DOS" (prononcez dosse= "DISC OPERATING SYSTEM"), dit "CPM", largement utilisé, et dont s'est copieusement inspiré IBM en créant son "MS/DOS" pour faire son "IBM PC". Une future formation sur IBM PC serait alors très rapide en venant de l'AMSTRAD : presque les mêmes Basic, et deux DOS ayant les mêmes "esprits" et souvent les mêmes syntaxes... Par contre, aucun rapport avec l'antique DOS des APPLE II. Qui c'est qui a dit "tant mieux" ?
Revenons plus en détail sur certains de ces points.

QU'EST-CE QUE LE DOS ?

C'est le "système d'exploitation", une sorte de langage, disons le "Basic du Drive". Ici, nous avons deux DOS, le CPM tout puissant et normalisé, et "l'AMSDOS". En effet, le fabricant a pris conscience que ce lecteur de disquettes s'adaptait à un micro-ordinateur conçu à l'origine pour magnétophone, et cet "AMSDOS" est un langage de dialogue entre le Basic résident et le CPM. Grâce à lui, nous disposons de fonctions nouvelles, telle la copie d'une cassette sur disquette, ou l'inverse (logiciels non protégés ou déplombés). Toutes les commandes Basic concernant le magnétophone telles que LOAD, SAVE, OPENIN, CAT, etc. (à l'exception de SPEED WRITE) s'appliquent directement au DRIVE.
Les différents sous-programmes CPM présents sur la disquette système (la précieuse) nous permettent une foule de charcutages de fichiers : copie de fichiers, copie de toute la face de disquette, changer les noms de programmes, effacer tel nom ou telle catégorie de fichiers, etc.
Dès que le DRIVE est branché au CPC 464, ce dernier oublie le magnétophone et ne s'adresse qu'à la disquette. Pour revenir au magnétophone, il suffit d'entrer ITAPE, et plus tard IDISC pour revenir au DRIVE ; ce sont là des commandes AMSDOS qui s'ajoutent donc aux commandes Basic existantes.

L'ACCES DIRECT

Pour vous démontrer les avantages, nous prendrons l'exemple des "gestionnaires de fichiers" :
Si le programme est proposé en cassette, il faut **tout le charger en RAM,** et il ne reste à peine que 20 kilo-octets pour notre fichier, donc 200 à 300 fiches maximum. Si ce logiciel était plus sophistiqué (nombreuses options), on aurait encore moins de place pour nos fiches !
En revanche, une bon logiciel sur disquette peut représenter un total de 70 k-octets et nous laisser 35 k-octets disponibles ! Comment cela ?
C'est très simple :
Le logiciel est décomposé en une douzaine de sous-programmes sur la même face. Au départ, on ne charge que le menu, disons 2000 octets. Vous choisissez l'option saisie, il charge le sous-programme SAISIE qui écrase MENU et vous remplissez votre fichier qui reste en RAM.
Option tri : charge le TRI qui vient écraser SAISIE, et ainsi de suite. Vue l'astuce ?
Mieux, en option "Recherche sous plusieurs critères", on peut demander d'explorer des fichiers énormes enregistrés sur disquette, fiche par fiche, en ne conservant en RAM que celles qui répondent aux critères demandés.
NOTE : Ces progiciels (logiciels professionnels) valent plus de 140 francs... Beaucoup plus...

OUI MAIS, ET LES INCONVENIENTS ?

Bien sûr il y en a, mineurs, mais ils existent.
— On perd 1284 octets de la mémoire disponible. Ainsi, lorsque l'on fait PRINT FRE(""), on obtient 42 249 octets disponibles au lieu de 43 533.
— Les noms de fichiers (ou de programmes) ne doivent pas excéder huit caractères, et sans espaces ni ponctuation.
— Après RUN", LOAD", SAVE", OPENIN" et OPENOUT", on est obligé de spécifier le **nom du fichier** ; et sans se tromper dans son orthographe ! C'est normal, puisque le DRIVE doit aller l'identifier dans le "catalogue" de la face de disquette en service.
— De ce fait, il faudra modifier la fin de certains programmes Basic qui enchaînent sur un autre. Exemple, le classique :

RUN"!

qui charge le programme qui suit sur la cassette en évitant l'inscription

"Loading MACHIN Block 1"

Donc, ici à remplacer par RUN "MACHIN", voire changer le nom du programme s'il est trop long ou comporte des espaces.
A noter au passage que vous n'avez plus à faire plusieurs sauvegardes d'un programme en cours d'écriture. Vous sauvez, par exemple, votre programme TRUC inachevé. Au catalogue, vous aurez en fait :

TRUC.BAS 2K

Le .BAS indique que c'est du Basic. Plus tard, nouvelle sauvegarde de travail par le même SAVE"TRUC" :
Au catalogue, on aura :

TRUC.BAK 2K
TRUC.BAS 4K

Le .BAK, c'est la version **précédente** qui n'a pas été écrasée. Seule son "extension de nom" (ou "type") a été modifiée.
Lorsque vous ferez LOAD"TRUC" ou RUN"TRUC", c'est TRUC.BAS qui sera exécuté.

PEUT-ON ENREGISTRER DES LOGICIELS DE JEUX ?

Depuis le temps que vous vouliez poser cette question...
Directement, non. Il faut des logiciels de déplombage pour charger en RAM et transférer sur disquettes (durs à trouver). Tous les logiciels de jeux, ou presque, sont une série de programmes en Basic et en binaire

(langage machine). Or, ces derniers posent parfois des problèmes d'emplacements mémoire.
C'est un travail assez long mais payant, car avoir ses cassettes préférées sur une même disquette et disponibles chacune après moins de dix seocndes, au lieu de cinq à dix minutes...
Ce n'est pas toujours facile, voire impossible. Disons que le pourcentage d'échecs se situe vers 25 %. Chose curieuse, parmi les 75 % de transferts réussis, l'opération inverse, disquette vers cassette, heureusement sans intérêt, se solde très souvent par un échec. Bien fait pour les pirates !

CPC 464 OU 664 ?

L'avènement du 664 a conduit à une baisse de prix du 464 et du DDI chez de nombreux détaillants, ce qui fait un prix d'ensemble à peine supérieur à 6500 F ; et là, on est complet. A partir du 664, il faudra accoller un **bon** magnétophone, plus encombrant que le DDI. Le prix global est alors sensiblement le même...
Si, par la suite, on s'offre un second DRIVE, on le pose SUR le DDI (pas plus encombrant) ou A COTE du 664 (plus d'encombrement).
Il n'est pas du tout évident de trancher en faveur du 464 ou du 664 ; c'est alors une affaire de cas personnels bien précis.

EN CONCLUSION

Une phrase classique résume tout : c'est mieux, mais c'est plus cher...
C'est très pratique, très agréable, mais pas indispensable. Car, mis à part certains logiciels professionnels, paraît-il en cours d'édition, le DRIVE n'apporte pas de "puissance" supplémentaire au système ; c'est comme comparer une chignole à main et une perceuse électrique : elles font exactement le même travail, mais avec des temps très différents (que le constructeur nous pardonne cette image !).
Comme il s'agit d'un investissement (éventuel) important, il était nécessaire de savoir ce qu'il apporterait ou non.

3

FONCTION CIRCLE

Eddy DUTERTRE

L'AMSTRAD a, on le sait, de bonnes capacités graphiques, mais la fonction CIRCLE manque à l'appel. Pour pallier ce défaut, nous vous proposons ci-dessous un petit programme en langage machine qui vous permettra d'obtenir cette fonction directement par son nom à partir du Basic. Pourquoi en langage machine plutôt qu'en Basic ? Tout simplement pour avoir une grande rapidité d'exécution.

FONCTIONNEMENT DU PROGRAMME

Pour éviter d'utiliser les routines SIN et COS de la ROM du CPC, nous avons opté pour l'implantation en mémoire d'une table de Sinus et de Cosinus d'angles compris entre 0 et 2 π (360°). Bien sûr, il n'était pas question de travailler alors avec des valeurs à virgule flottante, aussi, afin d'obtenir des nombres entiers, chaque valeur des tables a été multipliée par 127 (7 FH) et séparée de sa partie décimale. On obtient ainsi deux tables dont les données, comprises entre 0 et 127, correspondent en réalité à 1 – cos et 1 – sin, ceci afin de s'affranchir des valeurs négatives. Tout cela reviendra en fait à ne prendre en considération que deux chiffres après la virgule dans la table de sin et de cos. La suite devient alors très simple et se résume à de simples multiplications et divisions :
Connaissant les coordonnées X et Y du centre du cercle à dessiner ainsi que son rayon, on peut écrire, en partant du point A et en faisant varier l'angle de 0 à 2 π, que les coordonnées de chaque point du périmètre du cercle ont pour valeur :

$XD = X + R - R(1 - \cos \alpha) =$
$X + R \cos \alpha$
$YD = Y - R + R(1 - \sin \alpha) =$
$Y - R \sin \alpha$

avec :

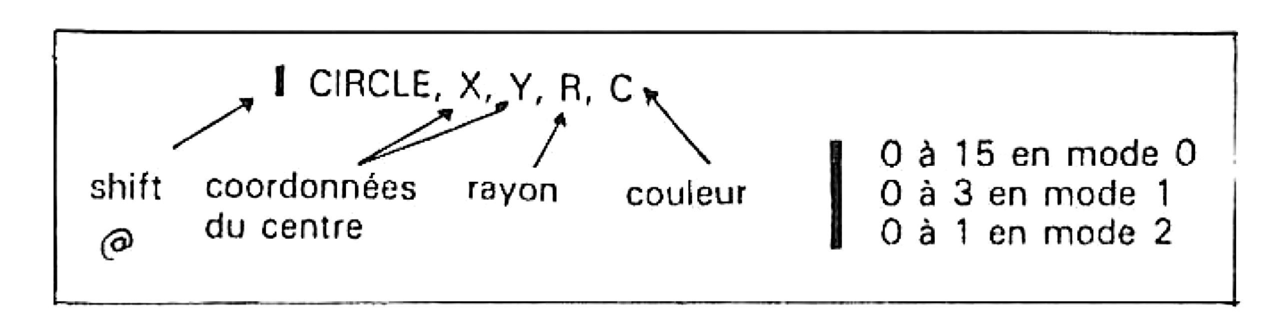
$XR = X + R$
$YR = Y - R$

et

$1 - \cos \alpha$
$1 - \sin \alpha$

contenus dans les tables.

$XD = XR - R(1 - \cos \alpha)$
$YD = YR + R(1 - \sin \alpha)$



UTILISATION DU PROGRAMME

Après avoir tapé soigneusement le programme et surtout les lignes de DATA, il suffit de faire RUN pour l'implanter en mémoire. Si tout se passe bien (aucune erreur), un cercle devra se dessiner au bout de quelques secondes.
La fonction CIRCLE sera alors disponible en faisant :

CIRCLE, X, Y, R, C

shift

coordonnées
du centre

rayon

couleur

0 à 15 en mode 0
0 à 3 en mode 1
0 à 1 en mode 2

Notez qu'il est alors possible de sauvegarder uniquement la routine en langage machine en faisant :

SAVE "CIRCLE",B,&A000,&1D4

Pour la recharger et la valider avant utilisation, il faudra faire :

MEMORY &9FFF
LOAD "CIRCLE"
CALL &A010

Dès lors, son utilisation sera possible en Basic par le mot CIRCLE précédé, ne l'oubliez pas, de " " (shift@).

LE PROGRAMME

Dans le souci de satisfaire les lecteurs qui ne disposent pas de matériel de reproduction, nous sommes en mesure de vous procurer les mylars de ces circuits imprimés qui sont disponibles à la rédaction contre un chèque de 70 francs. Les abonnés de CPC pourront les obtenir pour 50 francs en joignant à leur commande une étiquette d'abonné.

```
10 DATA C8,00,C8,00,64,00,02,2C,01,64
20 DATA 00,F9,00,64,00,00,01,19,A0,21
30 DATA 25,A0,C3,D1,BC,1E,A0,C3,29,A0
40 DATA 43,49,52,43,4C,C5,00,FC,A6,19
50 DATA A0,FE,04,C0,DD,66,07,DD,6E,06
60 DATA 22,00,A0,DD,66,05,DD,6E,04,22
70 DATA 02,A0,DD,66,03,DD,6E,02,22,04
80 DATA A0,DD,66,01,DD,6E,00,22,06,A0
90 DATA 3A,06,A0,CD,DE,BB,2A,00,A0,ED
100 DATA 5B,04,A0,19,22,07,A0,A7,2A,02
110 DATA A0,ED,5B,04,A0,ED,52,22,09,A0
120 DATA ED,5B,07,A0,2A,02,A0,CD,EA,BB
130 DATA DD,21,09,A1,FD,21,6F,A1,06,66
140 DATA C5,ED,5B,04,A0,DD,7E,00,0E,00
150 DATA CD,D0,A0,16,00,1E,7F,7C,4D,CD
160 DATA E1,A0,50,59,2A,07,A0,A7,ED,52
170 DATA 22,0B,A0,ED,5B,04,A0,FD,7E,00
180 DATA 0E,00,CD,D0,A0,16,00,1E,7F,7C
190 DATA 4D,CD,E1,A0,50,59,2A,09,A0,19
200 DATA 22,0D,A0,ED,5B,0B,A0,CD,F6,BB
210 DATA DD,23,FD,23,C1,10,B3,C9,06,10
220 DATA 21,00,00,CB,39,1F,30,01,19,EB
230 DATA 29,EB,10,F5,C9,21,00,00,06,10
240 DATA CB,11,17,ED,6A,ED,52,3F,30,11
250 DATA 10,F4,C3,03,A1,CB,11,17,ED,6A
260 DATA A7,ED,5A,38,F1,28,EC,10,F2,CB
270 DATA 11,17,19,47,C9,00,00,01,02,04
280 DATA 06,09,0C,10,14,18,1D,22,28,2E
290 DATA 34,3B,42,49,50,58,5F,67,6F,77
300 DATA 7F,87,8F,97,9F,A6,AE,B5,BC,C3
310 DATA CA,D0,D6,DC,E1,E6,EA,EE,F2,F5
320 DATA F8,FA,FC,FD,FE,FE,FE,FD,FC,FA
330 DATA F8,F5,F2,EE,EA,E6,E1,DC,D6,D0
340 DATA CA,C3,BC,B5,AE,A6,9F,97,8F,87
350 DATA 7F,77,6F,67,5F,58,50,49,42,3B
360 DATA 34,2E,28,22,1D,18,14,10,0C,09
370 DATA 06,04,02,01,00,00,00,7F,77,6F
380 DATA 67,5F,58,50,49,42,3B,34,2E,28
390 DATA 22,1D,18,14,10,0C,09,06,04,02
400 DATA 01,00,00,00,01,02,04,06,09,0C
410 DATA 10,14,18,1D,22,28,2E,34,3B,42
420 DATA 49,50,58,5F,67,6F,77,7F,87,8F
```

```
430 DATA 97,9F,A6,AE,B5,BC,C3,CA,D0,D6
440 DATA DC,E1,E6,EA,EE,F2,F5,F8,FA,FC
450 DATA FD,FE,FE,FE,FD,FC,FA,F8,F5,F2
460 DATA EE,EA,E6,E1,DC,D6,D0,CA,C3,BC
470 DATA B5,AE,A6,9F,97,8F,87,7F,77,XX
480 L=&A000
490 READ A$:IF A$="XX"THEN 510
500 A=VAL("&"+A$):POKE L,A:L=L+1:GOTO 490
510 CALL &A010 ' Implantation du RSX
520 CLS:|CIRCLE,319,200,150,1
530 DELETE -530
```

TRUCS ET ASTUCES

Raymond STOFFEL — 68 RIXHEIM

Ayant acheté un AMSTRAD CPC 664, j'ai été très intéressé par votre programme Mirage paru dans le numéro 1 de votre revue. Malheureusement, la compatibilité du 664 avec le 464 n'étant que ce qu'elle est, le programme en question ne fonctionnait pas correctement.
En fait, les adresses des pointeurs Basic ne sont pas identiques sur les deux ordinateurs.
Voici les correspondances qui permettront à vos lecteurs de modifier les deux programmes en question :

664	464
AE66-AE67	AE83-AE84
AE68-AE69	AE85-AE86
AE6A-AE6B	AE87-AE88

Par ailleurs, la modification des pointeurs Basic étant également responsable du plantage du programme de jeu SORCERY, il convient de procéder ainsi :

POKE &AE81,&6F:
POKE &AE82,&1

et de lancer le jeu normalement en appuyant sur les touches control et enter.

2

INITIATION A L'ASSEMBLEUR

Denis BOURQUIN

A partir de ce numéro, nous allons commencer une série d'articles d'initiation à l'assembleur. Abondamment traité certes, nous avons cependant pensé qu'il était nécessaire que ce sujet figure dans cet ouvrage. Le microprocesseur du CPC étant un Z80 de Zilog, nous traiterons donc plus particulièrement de l'assembleur Z80.

Dans cette première partie, nous ne ferons pas d'assembleur, mais nous essayerons d'introduire un certain nombre de notions indispensables à la bonne compréhension de la suite. La programmation en assembleur nécessite de posséder quelques connaissances (matériel et logiciel) que nous allons passer en revue.
L'AMSTRAD est construit autour du microprocesseur Z80 de ZILOG, ce microprocesseur est dit "8 bits" car il possède un bus de données 8 bits. Nous allons donner quelques explications pour les débutants.
Le microprocesseur dialogue avec les circuits qui l'entourent au moyen de trois bus (un bus étant une voie de dialogue à plusieurs liaisons), un bus de données de 8 bits (8 liaisons), un bus d'adresse sur 16 bits (16 liaisons) et un bus de commande. Le rôle du bus d'adresse est de fournir l'adresse du périphérique avec lequel le microprocesseur veut dialoguer. Cette commande correspond à la combinaison des 16 liaisons d'adresse. Le bus de commande permet au microprocesseur d'indiquer le sens du dialogue (lecture ou écriture), le type de périphérique adressé (mémoire, entrée-sortie), d'indiquer son état aux périphériques ou de recevoir des acquittements des périphériques.
Le plus important des circuits périphériques du microprocesseur sera la mémoire. Sans elle, il ne peut rien faire, c'est dans cette mémoire que seront stockés les programmes et les données. Elles est architecturée en "cases" de 8 bits appelées "octets". Le Z80 ayant un bus d'adresse de 16 bits, ils ne pourra donc adresser instantanément qu'une case mémoire choisie parmi 2↑16, soit 65536 cases.

Dans le Z80, comme dans un certain nombre d'autres microprocesseurs, nous distinguons, en plus de la mémoire, un autre espace adressable : celui des poids faibles du bus d'adresse ; il est donc de 2↑8, soit 256 cases, ces cases sont plus couramment appelées ports. Par exemple, la sortie imprimante du CPC est dans cet espace, de même que le générateur de sons, le contrôleur d'écran, le clavier, le contrôleur de disque. Mais vous aurez remarqué, dans votre manuel du CPC, que les adresses de l'espace entrée-sortie ne sont pas données sur 8 bits mais sur 16 (utilisation en Basic de l'instruction INP pour lire le contenu d'un port et de l'instruction OUT pour écrire dans un port). En effet, le décodage des adresses des entrées-sorties du CPC utilise une des particularités du Z80 : en mode d'adressage indirect des entrées-sorties par le registre C, le microprocesseur met sur les bits de poids forts du bus adresse le contenu du registre B, mais il est peut-être prématuré de parler de cette particularité ; aussi, nous y reviendrons quand nous analyserons les entrées-sorties en assembleur.
La mémoire du CPC est de deux types : de la ROM qui est une mémoire accessible uniquement en lecture et qui contient un programme figé par le constructeur. C'est dans cette ROM que nous trouvons l'interpréteur Basic du CPC. Elle est subdivisée en deux sections de 16 kilo-octets chacune (1 kilo-octet = 1024 octets), la première est implantée à l'adresse 0 et va jusqu'à l'adresse 3FFFH (les adresses sont données ici en hexadécimal à 16↑0* 15+16↑1*15+16↑2*15+16↑3*3=16383). La seconde va de l'adresse C000H à l'adresse FFFFH.
Le deuxième type de mémoire du CPC est la RAM. Cette mémoire est accessible en écriture et en lecture, et est disponible pour l'utilisateur qui y logera ses programmes et les données de son application. Elle s'adresse sur 16 bits et fait 64 kilo-octets (65536 cases). Si nous faisons le compte de la mémoire du CPC, nous trouvons 96 kilo-octets ; il y a un truc ! Effectivement, le microprocesseur ne peut adresser avec ses 16 bits d'adresse que 65536 octets, mais si nous utilisons un artifice distinguant différentes zones de 65536 octets, nous augmentons l'espace mémoire adressable. Cet artifice peut consister, par exemple, en écriture dans un port d'entrée-sortie d'une valeur correspondant à une des zones mémoires désirées, nous avons alors paginé ou segmenté la mémoire. Il suffit donc de préciser, avant d'y accéder, quel est le type de mémoire désiré. Le CPC procède par l'écriture dans un port, mais nous reparlerons de cette particularité plus tard.
Lorsque vous travaillez en Basic, vous n'êtes pas directement maître de la mémoire, l'interpréteur la gère à votre place. Vous n'y accédez directement que par les instructions POKE et PEEK. Rappelons que l'instruction PEEK (AD) retourne le contenu de la case mémoire d'adresse AD, et POKE AD,VAL écrit la valeur VAL dans la case d'adresse AD. En assembleur, c'est vous qui gérerez votre mémoire.
Nous allons arrêter là les considérations sur la mémoire, et nous allons nous intéresser d'un peu plus près au microprocesseur. Regardons l'architecture générale d'un micro. Cette architecture simplifiée est donnée par la figure 1.

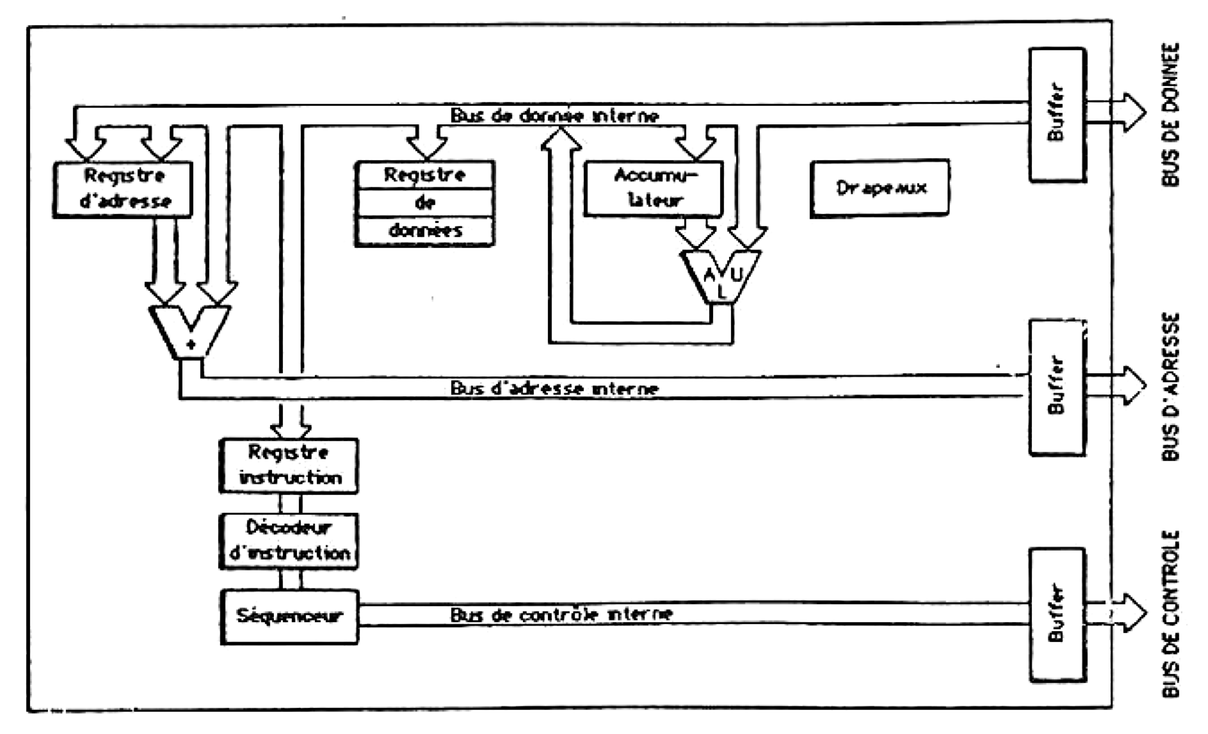


Figure 1

Nous trouvons, comme constituants principaux :

— Une unité arithmétique et logique ALU qui effectue les différentes opérations arithmétiques et logiques entre deux opérandes présentés sur ses entrées. Le plus souvent, l'un des opérandes est l'accumulateur ; cet accumulateur est un registre interne au microprocesseur (c'est une mémoire interne au microprocesseur). Le registre A, dans le Z80, a cette vocation. A cet accumulateur nous associerons le registre des drapeaux qui indique la manière dont s'est effectuée une opération arithmétique ou logique. Nous analyserons ultérieurement le contenu de ce registre des drapeaux.

— Des registres de données 8 bits. Ces registres sont utilisés comme mémoire temporaire des données dans les programmes. Il est toujours intéressant qu'un microprocesseur en possède plusieurs car le registre offre comme avantage, par rapport à la mémoire, d'être d'un accès plus rapide puisqu'interne au microprocesseur. Le Z80 possède deux banques de 6 registres. Ces registres sont repérés par une lettre et sont B, C, D, E , H et L pour la première banque, et B', C', D', E', H' et L' pour la deuxième banque. Une seule de ces banques et accessible à un instant donné, le passage d'une banque à l'autre s'effectuant par une seule instruction. Nous remarquons que le registre A correspond à la première banque et que son homologue, dans la deuxième banque, est le registre A'. Notons tout de suite que ces registres de 8 bits peuvent être regroupés deux par deux pour être utilisés en registres 16 bits : BC, DE, HL, ils peuvent servir de registres d'adresse.

— Des registres d'adresse 16 bits. Ce sont les registres IX et IY qui servent d'index pour l'adressage de zones mémoires. Nous reparlerons des différents types d'adressages. Le registre SP est le pointeur de pile. La pile, mais qu'est-ce que c'est ? Lorsque vous avez programmé en Basic, vous avez probablement utilisé des sous-programmees que vous avez appelés par GOSUB. A la fin du sous-programme vous aviez l'instruction RETURN. Oui, mais return où ? L'adresse de retour, l'interpréteur se l'était rangée dans un coin de la mémoire : la pile. La pile est une zone mémoire que vous réservez au microprocesseur, pour lui permettre de sauvegarder les adresses de retour au programme principal, lors des déroutements vers des sous-programmes ou vers des programmes d'interruption. Cette pile vous servira aussi, dans les programmes, à sauvegarder le contenu des registres. Sa structure est du type LIFO (last-in, first-out : dernier entré, premier sorti). La sauvegarde dans cette pile consiste finalement en des opérations d'empilage et de dépilage. Il nous reste encore un registre de 16 bits à voir ; le compteur ordinal : PC. Ce registre n'est pas manipulé directement par le programmeur, mais c'est lui qui pointe la mémoire pour aller chercher en séquence les différents codes opération et opérandes.

Le Z80 possède deux autres registres 8 bits : les registres I et R. Le registre I pointe la table des vecteurs d'interruption. Nous en reparlerons quand nous verrons les modes d'interruption du Z80. Le registre R fournit les adresses de raffraîchissement pour les mémoires dynamiques.

ROLE DE L'ASSEMBLEUR — 3

Après les quelques considérations matérielles exposées plus haut, nous allons parler de l'assembleur.
La première question que tout débutant se pose est : qu'est-ce qu'un assembleur ?
Un assembleur est un programme chargé de traduire un programme source en octets directement interprétable par le microprocesseur.
Le programme source est une suite d'instructions écrites avec leurs mnémoniques et n'est donc que du texte (suite de caractères ASCII).
L'utilisation d'un assembleur présente l'intérêt de ne pas avoir à rechercher la codification des opérations dans le langage binaire de la machine car, rappelons-le, le système de travail d'un microprocesseur est le système binaire. Les assembleurs offrent, en plus, d'autres services qui sont des aides à la programmation et à la relecture des programmes, comme l'utilisation des symboles pour repérer une case mémoire ou une zone mémoire, ou pour repérer un point de branchement dans un programme.
Dans un programme source, nous allons trouver des directives d'assemblage, des étiquettes, des réservations de zone mémoire, des lignes d'instruction.

Les directives sont des pseudo-instructions qui fournissent des indications au programme d'assemblage, mais ne provoquent aucune génération de code.
Nous allons regarder le format d'une ligne assembleur et expliquer ce que l'on peut y trouver.
Le format d'une ligne d'un programme source se décompose en quatre champs : les champs étiquettes, code opération, opérandes et commentaires.

Le cadre de la figure 1 donne l'alignement de ces champs et le caractère servant de séparateur. Analysons ces champs l'un après l'autre.
Un conseil avant tout : pour une bonne lisibilité de vos programmes, prenez l'habitude de cadrer vos lignes d'instructions de telle sorte que chaque champ occupe toujours le même nombre de caractères.

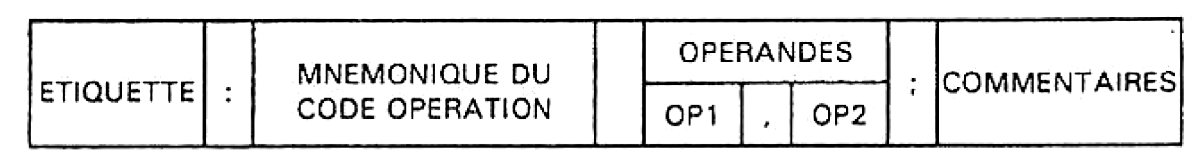


Figure 1

LE CHAMP ETIQUETTE

Il n'est pas toujours présent, il contient un symbole qui représentera l'adresse mémoire où sera rangé le premier octet de code de la ligne d'instruction. Les étiquettes sont utilisées pour représenter des points de branchement ou pour représenter le début d'un sous-programme.
Jusqu'à présent, nous n'avons pas parlé des symboles. Un symbole est une suite de caractères commençant toujours par une lettre, le nombre de caractères significatifs pris en compte dépend de l'assembleur.
Voici, par exemple, une suite d'étiquettes valides : CONIN, CHL1, RET45, IMPRIMANTE.
Et maintenant, voici une suite d'étiquettes incorrectes : 1BCL, HL, LD, END.
La première étiquette du deuxième exemple commence par un chiffre et non par une lettre ; la deuxième étiquette peut être confondue avec les registres du Z80, la troisième est une instruction Z80 et la quatrième une pseudo-instruction de l'assembleur. Le séparateur entre le champ étiquette et le champ opération peut être ":" ou un espace suivant le type d'assembleur utilisé.

LE CHAMP CODE OPERATION

Il contient le mnémonique du code opération, par exemple ADD. Le mnémonique du code opération est une abréviation du mot anglais désignant cette opération, par exemple, ADD est le mnémonique de l'addition, SUB le mnémonique de la soustraction, LD le mnémonique de

LOAD pour le chargement, mais nous les verrons lorsque nous analyserons les différentes instructions du Z80.
Dans le champ code opération, nous pourrons trouver des directives d'assemblages ; nous allons analyser les principales à la fin de ce chapitre.

LE CHAMP OPERANDE

Il contient 1, 2 ou pas d'opérande suivant le type d'opération réalisée ou suivant que le code opération sous-entend ou non la présence d'un ou deux opérandes. Lorsque nous aurons deux opérandes, ceux-ci seront séparés par une virgule.
Comme opérande, nous allons trouver :

Des opérandes registres
Rappelons qu'un registre est représenté par la lettre le désignant ; ce sont tous les registres 8 ou 16 bits du Z80 : A, B, C, D, E, H, L, BC, DE, HL, IX, IY, SP, I, R.

Des opérandes mémoires
Cet opérande mémoire pourra être un symbole que nous aurons déclaré par une directive assembleur, directive assembleur qui associera à ce symbole une case mémoire. Un tel opérande sous forme de symbole sera traduit par l'assembleur en une adresse absolue correspondant à la case mémoire associée. Nous trouverons donc aussi en opérande mémoire une référence directe à une adresse, par exemple, 16385, sera la case mémoire d'adresse 16385.
Lorsque nous adresserons une case mémoire en assembleur, la référence de cette case se fera toujours par l'intermédiaire de son symbole ou de son adresse entre parenthèses. Prenons tout de suite un exemple :

```
LD (CASE),A
```

L'instruction précédente est une instruction de chargement LOAD, elle prend le contenu du registre A et le met dans la case mémoire appelée ici CASE ; si cette case avait été réservée à l'adresse 16385, nous aurions pu écrire aussi :

```
LD (16385),A
```

Nous trouverons aussi en Z80 un autre type d'opérande mémoire : il correspond à un adressage indirect par registre. Nous n'avons plus, dans ce cas, directement l'adresse de la case mémoire, mais cette adresse se trouve dans un registre 16 bits. Revenons à l'exemple précédent. Pour stocker le contenu du registre A dans la case d'adresse 16385, nous avions déjà les deux méthodes suivantes :

```
ORG 16385
CASE DB
LD (CASE),A

LD (16385),A
```

Nous pourrions aussi faire :

```
LD HL,16385
LD (HL),A
```

Mais nous reparlerons de tous ces modes d'adressage dans l'analyse des instructions du Z80.

Des opérandes immédiats
Dans ce cas, l'opérande est directement une valeur que nous stockerons dans un registre ou une case mémoire. Notons que, dans ce cas, pour l'adressage en mémoire, seul l'adressage indirect par la paire de registres HL est valide.
Cette valeur immédiate peut être une valeur 8 bits si nous la chargeons dans un registre 8 bits, ou une valeur sur 16 bits si nous la stockons dans un registre de 16 bits.
Exemple : LD A,16 charge la valeur 16 dans le registre A.
Dans les exemples donnés jusqu'à présent, nous voyons apparaître une particularité de l'assembleur du Z80 qui veut que l'opérande destination soit placé avant l'opérande source, ce sera toujours le cas, quelle que soit l'opération (voir figure 2).

MNEMONIQUE OPERATION	OPERANDE DESTINATION		OPERANDE SOURCE

Figure 2

Champ opérande à un seul opérande
Il existe certaines instructions pour lesquelles un des opérandes est sous-entendu. Ceci, par exemple, est le cas de certaines opérations arithmétiques ou logiques comme la soustraction sans retenue qui s'effectuent toujours entre le registre A, qui a vocation d'accumulateur, et un autre opérande.
Exemple : SUB B soustrait le contenu du registre B du contenu du registre A ; le résultat est dans le registre A. Autre exemple : AND C qui est le ET logique entre le registre A et le registre C.

Champ opérande vide
Dans ce cas, il n'y a aucun opérande dans le champ opérande. Ceux-ci sont sous-entendus dans le mnémonique opération.

Exemple : RRA, mais nous les verrons de même avec l'analyse des instructions.

Champ opérande avec codes conditions

Nous avons là un autre cas d'opérande, la première partie du champ opérande contient une condition qui résultera d'une opération et la deuxième partie du champ sera une adresse absolue ou relative. Nous trouvons ce cas de champ d'opérande dans toutes les instructions de branchement.

```
JP Z,DEBUT
JR Z, DEBUT
```

Le format de l'instruction est celui de la figure 3.

Mnémonique du saut absolu relatif	Condition à réaliser	,	adresse absolue ou relative

Figure 3

Champ opérande avec opérateur

Le champ opérande peut contenir des opérandes plus complexes ; cette possibilité est offerte par les assembleurs comme une aide au programmeur. En effet, ceux-ci contiennent toujours un interpréteur d'opération plus ou moins rudimentaire.

Nous trouverons au moins comme opérateur les 4 opérations de base (* / + −).

Donnons tout de suite un exemple : nous pourrons écrire avec un assembleur :

```
LD HL,80*4+ECRAN
```

L'exemple donné donnera à l'assemblage si auparavant nous avons défini la directive :

```
ECRAN EQU 0C000H
```

l'équivalent de l'instruction :

```
LD HL,0C140H
```

Autre exemple plus simple : nous pouvons écrire :

```
LD A,16*4
```

ce qui est équivalent à :

LD A,64

Quel est l'intérêt de ce type d'écriture ? Le premier est de ne pas avoir à faire tous les calculs, et le deuxième de permettre une meilleure lisibilité des programmes.
Pour l'utilisation des opérateurs, il est impératif de se reporter à la documentation de l'assembleur utilisé, car ces derniers ne respectent pas toujours la hiérarchie des opérateurs ; par exemple l'assembleur ZEN évalue une expression strictement de gauche à droite et donc l'exemple précédent n'aurait pas donné le même résultat si on avait écrit :

LD HL,ECRAN + 80*4

Plus loin, nous essayerons de vous faire une comparaison entre les divers assembleurs en notre possession.

Le champ commentaire
Ce champ est toujours précédé d'un point-virgule. Nous y trouverons donc les commentaires du programmeur sur le fonctionnement de son programme ou pas de commentaire du tout. Encore une fois, nous vous donnerons un conseil : renseignez abondamment vos programmes, il vous sera plus facile de les reprendre quelques mois après.

4

Après avoir regardé le format d'une ligne assembleur, et avant de nous lancer dans les instructions du Z80, nous allons analyser quelques directives assembleurs rencontrées dans pratiquement tous les assembleurs Z80.

Les directives d'assemblage
Les directives d'assemblage sont des pseudo-instructions ; elles ne provoquent pas de génération de code objet, mais elles sont là uniquement pour permettre au programme d'assemblage de fonctionner normalement. Nous ne ferons pas le tour de toutes les directives, mais seulement des principales, nécessaires à l'écriture d'un programme ; nous introduirons les autres au fur et à mesure de l'analyse des instructions Z80 avec les exemples.

La directive ORG
La syntaxe de cette directive est de la forme ORG 4000H. Elle se compose de la commande ORG et d'une adresse définie sur 16 bits. Son rôle est d'indiquer au programme d'assemblage que ce qu'il va assembler commence à l'adresse indiquée, dans l'exemple à l'adresse 4000H (en hexadécimal). Comme ZEN est un des assembleurs les plus

répandus sur AMSTRAD, nous parlerons d'une directive particulière dont il est doté : la directive LOAD.

LOAD

ZEN est un assembleur qui assemble uniquement en mémoire ; il sera donc nécessaire de lui indiquer dans quelle zone mémoire il devra loger le code qu'il génère. Ce sera le rôle de la directive LOAD ; en l'absence de cette directive, il ne génère qu'un listing d'assemblage. Exemple : sous ZEN pour écrire un programme qui commencera à l'adresse 8000H et que l'on voudra exécuter ensuite ou sauver sur cassette, il faudra écrire :

```
ORG 8000H
LOAD 8000H
```

Cette directive LOAD n'existe pas sur les autres assembleurs qui génèrent le code objet soit dans un fichier disque, soit en mémoire lorsqu'ils possèdent l'option d'assemblage "en mémoire".

La directive EQU

Le rôle de cette directive est d'affecter une valeur à un symbole, symbole qui sera une constante ; cette fonction est l'équivalent du CONST en Pascal. Chaque fois que l'assembleur rencontrera, par la suite, ce symbole, il le remplacera par sa valeur.
Exemple :

```
CAS EQU 5
LD C,CAS
```

est équivalent à LD C,5 et l'assembleur génère le code correspondant à LD C,5.
Autre exemple :

```
MOTCAS EQU 0F610H
LD BC,MOTCAS
```

L'assembleur génère le code équivalent à LD BC,0F610H. Nous avons associé à MOTCAS la valeur hexadécimale 0F610H, et tout au long de notre programme, ce symbole vaudra 0F610H.
Normalement, dans les assembleurs, un symbole défini par la directive EQU ne peut pas être réaffecté ; nous ne pouvons trouver la directive "ETIQUETTE" EQU qu'une seule fois dans un programme.
Dans les exemples, nous avons écrit les constantes soit en décimal, soit en hexadécimal ; nous allons ouvrir une parenthèse sur la manière d'entrer une constante en assembleur.

LES CONSTANTES

Nous pouvons, dans une ligne instruction, écrire une valeur immédiate :
— en notation *décimale* : en général, l'écriture est la suivante : 10, 15, 0 ;
— en notation *hexadécimale* : dans ce cas, on indique que la base est différente en mettant la lettre correspondant à la base de numérotation à la fin de la constante ; la lettre pour la notation hexadécimale est la lettre H. La notation hexadécimale est la base 16, et nous avons donc 16 chiffres qui sont 0 à 9 et de A à F.

Les constantes octales
Comme pour la représentation hexadécimale, la représentation octale sera indiquée par une lettre, cette lettre est la lettre O après la constante. Exemple : 23O.
23O est équivalent à 19 en décimal et à 13H. Les chiffres dans cette représentation vont de 0 à 8.

Les constantes ASCII
Un caractère peut être représenté par un code international abondamment utilisé en informatique : le code ASCII. Par exemple, la lettre A est codée par le nombre 65 ou 41H en hexadécimal. Nous pourrons donc utiliser les lettres sous leur représentation ASCII en assembleur ; par exemple pour l'affichage d'un texte à l'écran, le code d'échange des caractères à afficher sera le code ASCII.
Exemple : en assembleur nous pourrons écrire :

```
LD A,'B'
```

ce qui est équivalent à :

```
LD A,66
```

car le code ASCII de 'B' est la valeur 66.
Exemples : 10H, 0FFH, 0C000H
Comme dans ce système de numérotation nous utilisons des chiffres qui sont représentés par des lettres, et pour éviter toute confusion avec des symboles, lorsque le premier chiffre d'une constante hexadécimale est une des lettres A à F, il est impératif de mettre devant un 0, sinon une erreur sera générée à l'assemblage.

Revenons à nos directives d'assemblage.
Dans un programme, nous traitons des données, il va donc être nécessaire de leur réserver de la place en mémoire. Nous avons, pour cela, un certain nombre de directives, ce sont les directives "Define".

Directive DB ou DEFB (define byte)
Le rôle de cette directive est de réserver un ou plusieurs octets en mémoire.
Exemple : MESSAGE DB 'BONJOUR' Le message est dans le champ étiquette, DB dans le champ opération, 'BONJOUR'.
Dans le champ opérande, cette ligne d'instructions réservera une zone mémoire ; dans la première case de cette zone, nous aurons la valeur 65, dans la deuxième la valeur 79, etc. Chacune de ces valeurs étant le code ASCII de la lettre correspondante : 65 est le code ASCII de B et 79 le code ASCII de O. Par la suite, nous ne parlerons plus de cases mémoire mais d'octets, car chacune de nos cases mémoire est un octet, comme nous l'avions expliqué précédemment.

Lorsque, dans la suite du programme, nous ferons référence à MESSAGE, ce sera à l'octet d'adresse correspondant à la case où se trouve la lettre B (valeur 65), c'est-à-dire si nous avons :

```
ORG 7000H
MESSAGE DB 'BONJOUR'
```

la lettre B se trouve à l'adresse 7000H, la lettre O à l'adresse 7001H, etc.
Autres exemples :
VAL1 DB 10
VAL2 DB 0F5H
TAB DB 10,11,12,14
Le troisième exemple a réservé 4 octets aux emplacements 10, 11, 12 et 14.

Directive DW, DEFW (define word)
Par cette directive, nous ne réservons plus un octet, mais un groupe de deux octets appelé mot (mot de 16 bits).
Exemple : MOT1 DW 0F542H
Nous avons, par cette directive, réservé deux octets en mémoire dans lesquels nous aurons mis les valeurs 42H pour le premier octet et F5H pour le second. Nous voyons ici une autre particularité du Z80 qui, pour stocker des mots en mémoire, stocke d'abord l'octet de poids faible, puis l'octet de poids fort.
Dans la suite du programme, si nous écrivons LD HL,(MOT1), nous mettrions dans la paire de registre HL la valeur F542H, le registre H à la suite de cette instruction contenant 0F5H et le registre L contenant 42H.
Nous pouvons trouver aussi : BRANCH DW TABLE avec TABLE qui est une étiquette de programme et dans ce cas, BRANCH est initialisé à la valeur de l'adresse correspondant à TABLE.

Directive DS ou DEFS (define storage)
La syntaxe de cette directive est la suivante :

```
BUFFER DS 128
```

Dans cet exemple, nous avons réservé une zone mémoire de 128 octets que nous avons appelée BUFFER.
Cette directive n'affecte pas une valeur à un octet, mais réserve un certain nombre d'octets ; nous trouvons derrière DS ou DEFS le nombre d'octets réservés.
La syntaxe employée pour les différentes directives utilisait l'espace comme séparateur. Les utilisateurs de ZEN auront remarqué que leur assembleur n'utilisait pas la même syntaxe.
Comme il a été dit plus haut, ZEN utilise les deux points comme séparateur du champ étiquette et du champ instruction ; nous aurons donc avec ZEN l'écriture suivante pour les directives DEFINE :

```
MOT1:DW 0F542H
MESSAGE:DB 'BONJOUR
BUFFER:DS 128
```

3

LE BUG DE L'OPENOUT

Michel ARCHAMBAULT

Toute ROM a ses petits bugs, et le Basic du CPC 464 n'échappe pas à cette triste règle. Or, ce bug peu génant avec cassettes devient un lourd handicap si l'on se sert d'un lecteur de disquettes. Voici les faits :

```
6100 INPUT "NOM du FICHIER", FICH$
6110 OPENOUT FICH$
6120 WRITE #9, etc.
6200 CLOSEOUT
```

S'il y a peu de variables en RAM, si le fichier FICH$ est assez petit (moins de 4000 octets environ), il n'y aura aucune anomalie, c'est-à-dire qu'en faisant CAT, nous aurons bien le nom du fichier FICH$.
A présent, le fichier fait déjà 5 ko, sauvegarde CAT. Oh ! surprise ! pas de FICH$ au catalogue (ou en-tête de fichier) mais un nom bizarre constitué par un fragment des données de FICH$! Autre inconvénient : le programme se bloque de nombreuses secondes avant et après l'enregistrement sur cassette ou disquette. Idem d'ailleurs au rechargement par OPENIN.
Lorsque l'on utilise la magnétophone, on ne s'aperçoit guère de la détérioration du nom du fichier, car on a, par exemple :

```
9100 OPENIN " "
```

ou mieux encore OPENIN"!" qui supprime les messages à l'écran "LOADING MACHIN Block 1".
Le programme charge donc le premier fichier qu'il rencontre sur la cassette en lecture. En revanche, une telle écriture ferait planter avec un "Bad Command" s'il doit lire sur une disquette ! Il faut alors que OPENIN soit suivi du vrai nom du fichier, FICH$; et si celui-ci a été "détérioré" lors de son enregistrement, il y aura plantage. Motif : "nom de fichier inconnu"... Plutôt gênant !

Voici d'abord le remède (trouvé non sans peine...) et ensuite l'explication.

LE REMEDE

Au début d'un programme Basic susceptible d'enregistrer ou de lire un long fichier, insérez les lignes suivantes :

```
50 OPENOUT "BIDON"
60 MEMORY HIMEM-1:
   CLOSEOUT
```

Très important :
— les définitions de variables (ex.: N=15) ou de DIM (ex.: DIM AD$(200,5) seront placées APRES ces deux lignes ;
— en revanche, un SYMBOL AFTER doit être logé AVANT ces lignes.
Un exemple très concret : notre programme LABELMATIC publié dans CPC n° 1, page 21 sera amélioré de la façon suivante :

— efface la ligne 300 (=GOSUB 51000)
— 30 GOSUB 51000
— écrire les lignes 50 et 60

Le GOSUB 51000 exécute un petit sous-programme qui établit les caractères AZERTY accentués, et il commence par un SYMBOL AFTER... On a donc déplacé ce GOSUB de la ligne 300 à la ligne 30. Sans cette précaution, on aurait droit à un "Memory full" (mémoire pleine).

EXPLICATION DU PHENOMENE

Le programme Basic se loge dans le "bas" de la mémoire, tandis que les variables s'empilent en "haut", juste avant la zone RAM réservée à la mémoire d'écran. Cette limite supérieure de la mémoire a pour adresse HIMEM (=HIGH MEMORY). On peut l'abaisser par la commande MEMORY.
Or, voilà que le programme rencontre la commande OPENOUT, et voici la farce que cela provoque :
Le HIMEM s'abaisse automatiquement de 4096 octets, soit 4 kilo-octets, l'emplacement pour loger deux "blocks". Donc, tout le paquet des variables qui se trouvaient juste en-dessous, voient leurs adresses abaissées de 4096, et si elles sont nombreuses, ce relogement peut prendre du temps...
Si on avait demandé OPENOUT "AGENDA", le fichier serait enregistré sans problème sous ce nom fixe et programmé. En revanche, si c'est

OPENOUT FICH$, il doit aller chercher FICH$ dans le bloc qu'il a déplacé (ou qu'il n'a pas fini de déplacer ?). Et c'est là que se manifeste le bug de la ROM AMSTRAD CPC 464 : si le bloc de variables est trop important, il "tape à côté" dans la mémoire et nous ramène autre chose en guise de FICH$.
Arrive le CLOSEOUT, le HIMEM reprend son ancienne valeur, nouveau déplacement de la mémoire, mais vers le haut, tout aussi lent qu'à la descente...
Avec OPENIN FICH$, c'est tout aussi long, mais il n'y a pas d'erreur pour retrouver la bonne valeur. Le bug ne concerne donc que OPENOUT.

A présent, expliquons le remède :
Cet OPENOUT "BIDON" de la ligne 50 a pour seul but de faire abaisser l'HIMEM ; comme il n'y a encore rien "en-dessous", c'est instantané. C'est alors l'occasion de figer définitivement le sommet de la mémoire disponible par MEMORY HIMEM – 1, même après le CLOSEOUT. On laisse donc vide cet espace de 4 ko pour que les OPENOUT et OPENIN futurs ne fassent plus ces "coups d'accordéon" sur le bloc des variables, lequel restera fixe. Plus d'erreurs de noms de fichiers, plus d'attentes avant et après les enregistrements ou lectures des fichiers.

4

LES FONDATIONS D'UN PROGRAMME BASIC

Michel ARCHAMBAULT

Les premiers programmes Basic d'un débutant commencent presque toujours par PRINT ou INPUT. Vingt ou trente lignes plus loin, on fait RUN, et tout s'écroule. De même que l'on ne construit pas une maison sur un sol meuble, il est suicidaire de se lancer dans un programme sans prendre des précautions dès le départ. Il ne s'agit que de quelques bonnes habitudes à prendre.

La première chose à faire, c'est sur papier, c'est le plan, l'itinéraire, la stratégie à suivre portant le nom pompeux d'ORGANIGRAMME; même si ce ne sont que quelques notes griffonnées sur une demi feuille. Nous n'en parlerons pas, car aujourd'hui on va rester très terre-à-terre.

LES NUMEROS DE LIGNES

Les numéros autorisés vont de 1 à 65535 et chacun d'entre eux a le même encombrement mémoire, deux octets seulement. D'autre part, il est très important de savoir que le "pas" (l'écart) entre deux numéros de lignes qui se suivent n'a absolument aucune incidence sur la vitesse d'exécution du programme. Par exemple, un programme de six lignes numérotées de 1 à 6 (donc au pas de 1), n'ira absolument pas plus vite que s'il était au pas de 10 000 entre 10 000 et 60 000. Donc n'ayez aucune hésitation pour espacer vos numéros.
— Dans une suite "naturelle" de lignes, le pas minimum est de 10.
— Dès la moindre transition, passez à la centaine suivante, exemple la suite 600, 610, 620, 700 (ou carrément 800).
— Dès que l'on attaque une autre partie du programme, on passe au millier suivant. Exemple : 1220, 1230, 2000, 2010.
— Chaque ligne en mille est un REM, exemple 3000' CALCULS DES MOYENNES.

Ne croyez surtout pas que c'est "pour faire joli" ! Ce sont plutôt des ruses pour subir sans douleur les deux grandes lois de l'informatique ; à savoir :

1 — UN PROGRAMME NE FONCTIONNE JAMAIS DU PREMIER COUP

2 — UN PROGRAMME PARFAIT N'EST JAMAIS DÉFINITIF.

Les principaux avantages sont les suivants :
— Grâce à ce saut à la centaine ou au millier suivant, nous pourrons très facilement ajouter plusieurs lignes à tel passage. Si vous pressentez une future mise au point laborieuse, sautez **plusieurs** centaines ou milliers. C'est gratuit.
— La localisation rapide. Exemple : la présentation de telle page d'écran ne vous plaît guère, vous savez que c'est la "partie 8000". Alors faites : LIST 8000-8900 (même si la 8900 n'existe pas). Cinq secondes plus tard, vous avez identifié à l'écran la fameuse ligne à modifier. Si tout le listing était continu au pas de dix et sans REM, il faudrait compter au moins cinq à dix minutes pour retrouver cette ligne...

ANNONCEZ LA COULEUR

Avant qu'un programme démarre, il faut fixer de nombreux paramètres, même s'ils ne servent que vers la fin. Autrement dit, on "jette les bases".
— Tout d'abord, en ligne 10, un REM où se trouvent le **nom du programme** et sa date. Pas celle de sa création mais celle de sa **dernière modification** ; ceci afin de pouvoir identifier la version la plus récente...
— Fixer les variables entières par DEFINT. S'il n'y aura aucun nombre décimal ou supérieur à 32 767 ou inférieur à - 32 768, écrire 50 DEFINIT A-Z.
Sinon (par convention) DEFINT I-N. Ce qui veut dire que toutes les variables numériques dont le nom **commence** par les lettres I, J, K, L, M ou N seront des nombres entiers. On peut aussi écrire DEFINT A,B, I-P, Z. Deux énormes avantages : chaque variable entière occupe deux octets au lieu de cinq. Les boucles FORNEXT dont l'indice est entier,vont deux fois plus vite !
NOTE : La "convention" entiers de I à N vient du langage FORTRAN car I et N sont les premières lettres du mot "INTEGER" signifiant entier en anglais. D'où l'usage courant de ces lettres comme indices de comptage (FOR I = 1 TO..., N = N + 1, etc.).
— Confirmer les paramètres d'écran même s'il s'agit de valeurs "normales", ce afin de ne pas hériter de celles du programme précédant (qui restent actives même après un NEW). Exemple :
60 INK 0,1 : INK1, 24 : INK2, 20 : INK3,6 : BORDER 1 70 PAPERO : PEN 1 : MODE 1 : CLS : ORIGINO,0.
— Tous les DIM sur la même ligne. Exemple :
100 DIM AC$ (24), NB (120), RESULT (12,3), REC$ (64)

Mettez cette suite en ligne 100, parce que c'est facile à retrouver. Les avantages :

- On n'a pas à réécrire "DIM" à chaque fois
- En cours de programmation, si on veut modifier ou ajouter un nouveau tableau DIM, on n'a qu'à faire EDIT 100.
- Tous les DIM étant définis en **début** du programme, on est certain qu'un GOTO ou un GOSUB ne viendra pas "repasser" dessus, provoquant ainsi un plantage avec "Redim Array Error" (= tableau redimensionné).

LE MENU-CARREFOUR

Avez vous remarqué que tous les logiciels du commerce, jeux ou utilitaires, on un "menu-carrefour" ? Une page d'écran qui propose les options principales chacune d'entre elles peut aiguiller sur d'autres menus ; mais en fin d'exécution (ou de "GAME OVER"), on revient à ce même point de départ, d'où on peut repartir vers une autre voie. C'est l'architecture type de tout programme informatique (même en "grosse informatique"), c'est en quelque sorte le retour systématique à la table des matières d'un manuel : on va à tel chapitre ou on décide de fermer le livre.
Nous ferons de même, c'est facile, mais surtout si pratique : une "bonne" habitude consiste à ce que la première ligne de cette étape menu porte le numéro 1000. Toujours la même raison, c'est facile à retenir tout comme 10 = titre + date et 100 = DIM. Avantage : en cas d'arrêt du programme, soit volontaire par la touche ESC ou plantage imprévu, il est facile de "reprendre" la main" en conservant les variables en cours en tapant :

GOTO 1000, et Enter

Lorsque l'on doit intervenir sur un programme Basic vieux de plusieurs années, c'est bougrement pratique, cet "universel" GOTO 1000 !

LES MODULES OU SOUS-PROGRAMMES

A la suite du menu principal, on a les différents départs qui, eux aussi reviennent à la même ligne (sauf pour END), et que trouvons-nous sur cette ligne ? GOTO 1000.

En somme, le "moteur" du programme va des lignes 10 à 2000, ce qui peut faire une trentaine de lignes, que le programme fasse deux ou vingt kilo-octets ! Ce qui change, c'est la taille et le nombre de ces modules commençant en ligne 3000. Avantages :
Vous avez conçu un petit programme qui marche bien, et tellement intéressant que vous décidez de l'étoffer, d'en faire un gros programme. Ne le retapez pas ! Ajoutez des lignes aux divers modules, ou refaites complètement tel module. Par exemple, le "bloc 8000" qui faisait 10 lignes va maintenant en faire 68 ! On peut créer de nouvelles options, donc de nouveaux sous-programmes, et ce **sans déranger le reste !**

QUELQUES REMARQUES

- Vous voulez un joli titre avec animation, son, etc. Vous ne pouvez pas le loger entre les lignes 500 et 990. Faites simplement :

```
500 'TITRE
510 GOSUB 20000
```

Et "là-bas", vous aurez toute la place nécessaire.

— Dans notre exemple, nous avons prévu les lignes 2010 à 2020 pour restituer les options par défaut que l'on a lors de la mise sous-tension (PAPER, PEN, INK, ORIGIN, etc.) ; seulement celles que l'on a dérangées en cours du programme. C'est remettre le micro dans l'état où on l'a trouvé. Un peu d'ordre évite bien des bizarreries sur le programme que l'on chargera après. Si vous vous êtes servi d'une imprimante, mettre **au début et à la fin** du programme :

```
PRINT#8,CHR$(27) ; "      " ;
```

qui vide le buffer des consignes de l'imprimante.

CONCLUSION

Toutes ces méthodes, ces "bonnes habitudes", n'ont vraiment rien de contraignant, et nous avons démontré tous les avantages que l'on pouvait tirer d'une architecture rationnelle sur des fondations solides.

4

REDEF'CAR

Stéphane CLOIREC

Vous qui avez tant peiné, armé de votre feuille de papier et de votre fidèle crayon, pour créer les caractères graphiques indispensables à vos programmes, ne désespérez plus !

REDEF'CAR va simplifier ce qui était autrefois une corvée, grâce aux différentes fonctions qui suivent :

— Vous manœuvrez le petit curseur dans la fenêtre d'édition grâce au pavé de curseur.

— Lorsque vous appuyez sur la barre d'espace, la case sur laquelle le curseur se trouve change de couleur (noire si elle était blanche et blanche si elle était noire).

— (E) Efface le caractère de travail.

— (I) Permet de faire une version "inversée" du caractère de travail (négatif de photo).

— (M) Comme miroir, produit une image retournée du caractère de travail par rapport à l'axe vertical.

— (D) Cette fonction vous permet de modifier ou de continuer un caractère déjà existant. Il vous faut rentrer les 8 valeurs du "SYMBOL", puis le programme vous rend la main.

— (S) Utilisant une routine en langage machine, ceci permet de conserver, sous la forme d'une ligne Basic (ex. : SYMBOL 140,1, 2, 4, 8, 16, 32, 64, 128), le caractère que vous venez de créer.

— (Q) Quitte le programme et vous permet de lister les caractères que vous avez mémorisés avec la fonction "SYMBOL" (S).

ATTENTION :

— Avant chaque utilisation du programme, resetez l'ordinateur par CTRL SHIFT ESC.

— Après avoir quitté le programme, et si vous avez conservé des caractères par "SYMBOL", n'oubliez pas de les sauvegarder pour un éventuel "MERGE".

— N'oubliez pas d'intégrer au programme utilisant les caractères graphiques l'ordre : SYMBOL AFTER 139.

— Il faut faire "RUN" après avoir quitté l'éditeur, pour pouvoir visualiser les symboles créés.

```
10 ' ###############
20 ' #             #
30 ' #  REDEF'CAR  #
40 ' #             #
50 ' #    LOADER   #
60 ' #             #
70 ' ###############
80 '
90 '
100 SYMBOL AFTER 129
110 MODE 1:SPEED INK 50,50
120 SYMBOL 133,195,231,52,24,44,231,195,0
130 SYMBOL 134,0,231,20,40,231,0,0,0
140 SYMBOL 139,4,8,60,102,126,96,60,0
150 INK 0,13:BORDER 13
160 INK 1,0:INK 2,24:INK 3,2,20
170 WINDOW #1,6.38,5,12:PAPER #1,1:CLS#1
180 WINDOW #1,5,37,4,11:PAPER #1,2:CLS#1
190 WINDOW #2,10,35,17,19:PAPER #2,1:CLS#2
200 WINDOW #2.9,34.16,18:PAPER #2.2:CLS#2
210 LOCATE #1.13,2:PRINT#1,"REDEF'CAR"
220 LOCATE #1.13,3:PRINT#1,STRING$(9,133)
230 LOCATE #1.4,6:PRINT#1."REDEFINISSEUR DE CARACTERES"
240 LOCATE #1.4,7:PRINT#1.STRING$(27,134)
250 LOCATE #2,3,2:PRINT #2.CHR$(164):"1985 St";CHR$(139);"phane CLOIREC"
260 PEN 3:LOCATE 12,24:PRINT"Chargement en cours."
270 FOR I=1 TO 2500:NEXT I
280 RUN"!REDEF1"
```

```
10 ' #################
20 ' #               #
30 ' #   REDEF'CAR   #
40 ' #               #
50 ' # (c) S.CLOIREC #
60 ' #               #
70 ' #################
80 '
90 '
```

```
100 INK 3,26:LOCATE 6,24:PEN 3:PRINT"Voulez-vous les instructions ?"
110 IF INKEY(34)=0 THEN GOSUB 1790 ELSE IF INKEY(46)<>0 THEN 110
120 '
130 '  FENETRES
140 '
150 INK 0,13:INK 1,0:INK 2.24:INK 3,24.13:MODE 1:BORDER 13
160 '
170 WINDOW #0,2.40.23.25:PAPER #0.1:CLS #0
180 WINDOW #1,2.11,2.11:PAPER #1.1:CLS #1
190 WINDOW #2.14,22,2.11:PAPER #2,1:CLS #2
200 WINDOW #3.25.40,2,11:PAPER #3,1:CLS #3
210 WINDOW #4.2.40,14.20:PAPER #4.1:CLS #4
220 WINDOW #0,1,39,22,24:PAPER #0,2:CLS #0
230 WINDOW #1,1,10,1,10:PAPER #1,2:CLS #1
240 WINDOW #2,13,21,1,10:PAPER #2,2:CLS #2
250 WINDOW #3,24,39,1,10:PAPER #3,2:CLS #3
260 WINDOW #4,1,39,13,19:PAPER #4,2:CLS #4
270 '
280 '  CARACTERES GRAPHIQUES
290 '
300 SYMBOL AFTER 128:DEFINT A-Z:DIM VLIG(8),CAR(8,8):SPEED INK 50,50
310 '
320 SYMBOL 129,255,129,129,129,129,129,129,255
330 SYMBOL 130,255,153,153,231,231,153,153,255
340 SYMBOL 131,231,231,231,24,24,231,231,231
350 SYMBOL 132,255,255,255,255,255,255,255,255
360 SYMBOL 133,195,231,52,24,44,231,195,0
370 SYMBOL 134,0,231,20,40,231,0,0,0
380 SYMBOL 135,0,4,70,123,123,70,4,0
390 SYMBOL 136,0,0,0,248,124,0,0,0
400 SYMBOL 137,244,135,245,7,120,96,28,252
410 SYMBOL 138,225,128,228,8,18,37,71,133
420 SYMBOL 139,4,8,60,102,126,96,60,0
430 '
440 '  INITIALISATION
450 '
460 KEY 138,"INK 1,0:INK 0,13:BORDER 13:PEN 1:PAPER 0:MODE 2:LIST"+CHR$(1
3)
470 PEN 1
480 LOCATE 2,2:PRINT CHR$(137);" REDEF'CAR ";CHR$(138);"  ";CHR$(164)+"19
85 St";CHR$(139);"phane CLOIREC":LOCATE 4,3:PRINT STRING$(9,134)
490 NUMLIG=1:NUMSYMB=140:ADR=&7000
```

```
500 GOSUB 1070
510 LOCATE #3,6,2:PRINT#3,"[MODE]":PRINT#3,SPC(5)+STRING$(6,133)
520 LOCATE #3,3,5:PRINT#3,"ZERO     UN"
530 LOCATE #3,3,6:PRINT#3,STRING$(4,134):SPC(5):STRING$(2,134)
540 LOCATE #4,16,2:PRINT#4,"COMMANDES":PRINT#4,SPC(15):STRING$(9,133)
550 LOCATE #4,1,5:PRINT#4,"  [D] MODIFIE  [E] EFFACE  [Q] QUITTE"
560 PRINT#4,"  [I] INVERSE  [M] MIROIR  [S] SYMBOL"
570 '
580 '   ROUTINE PRINCIPALE
590 '
600 CALL &BB18
610 IF INKEY(8)=0 THEN X=X-1 ELSE IF INKEY(1)=0 THEN X=X+1
620 IF INKEY(0)=0 THEN Y=Y-1 ELSE IF INKEY(2)=0 THEN Y=Y+1
630 IF X<XMIN THEN X=XMIN ELSE IF X>XMAX THEN X=XMAX
640 IF Y<YMIN THEN Y=YMIN ELSE IF Y>YMAX THEN Y=YMAX
650 LOCATE #1,A,B:IF CAR(A-1,B-1)=0 THEN PRINT #1,CHR$(129) ELSE PRINT #1
,CHR$(132)
660 LOCATE #1,X,Y:IF CAR(X-1,Y-1)=0 THEN PRINT #1,CHR$(130) ELSE PRINT #1
,CHR$(131)
670 LOCATE #1,X,Y:IF INKEY(47)<>0 THEN 700
680 IF CAR(X-1,Y-1)=0 THEN CAR(X-1,Y-1)=1:PRINT#1,CHR$(131):VLIG(Y-1)=VLI
G(Y-1)+2^(9-X):LOCATE#2,9-LEN(STR$(VLIG(Y-1))),Y:PRINT#2,STR$(VLIG(Y-1)):
PLOT 408+X*4,278+16-2*Y,1:PLOT 408+X*4+2,278+16-2*Y,1:PLOT 550+X*2,278+16
-2*Y,1:GOTO 700
690 CAR(X-1,Y-1)=0:PRINT#1,CHR$(130):VLIG(Y-1)=VLIG(Y-1)-2^(9-X):LOCATE#2
,6,Y:PRINT#2,"   ":LOCATE#2,9-LEN(STR$(VLIG(Y-1))),Y:PRINT#2,STR$(VLIG(Y-
1)):PLOT 408+X*4,278+16-2*Y,2:PLOT 408+X*4+2,278+16-2*Y,2:PLOT 550+X*2,27
8+16-2*Y,2
700 A=X:B=Y
710 IF INKEY(58)=0 THEN GOSUB 1070
720 IF INKEY(67)=0 THEN CLS:LOCATE 12,2:PRINT" FIN DE PROGRAMME.":GOSUB 1
640:MODE 2:CALL &B000:POKE &170+LENDAT+1,0:POKE &170+LENDAT+2,0:END
730 IF INKEY(35)=0 THEN GOSUB 780
740 IF INKEY(61)=0 THEN GOSUB 930
750 IF INKEY(38)=0 THEN GOSUB 1220
760 IF INKEY(60)=0 THEN GOSUB 1380
770 GOTO 600
780 '
790 '   ROUTINE  INVERSE'
800 '
810 FOR I=1 TO 8:VLIG(I)=0:NEXT I
820 FOR J=1 TO 8:FOR I=1 TO 8:IF CAR(I,J)=0 THEN CAR(I,J)=1 ELSE CAR(I,J)
```

```
=0
830 IF CAR(I,J)=1 THEN VLIG(J)=VLIG(J)+2^(8-I)
840 NEXT I,J
850 FOR J=1 TO 8:FOR I=1 TO 8
860 LOCATE #1,I+1,J+1:IF CAR(I,J)=1 THEN PRINT #1,CHR$(132) ELSE PRINT #1
,CHR$(129)
870 IF CAR(I,J)=1 THEN COUL=1 ELSE COUL=2
880 PLOT 408+(I+1)*4,278+16-2*(J+1),COUL:PLOT 408+(I+1)*4+2,278+16-2*(J+1
),COUL:PLOT 550+(I+1)*2,278+16-2*(J+1),COUL
890 NEXT I,J
900 IF FLAG=1 OR FL1=1 THEN RETURN
910 GOSUB 1190
920 RETURN
930 '
940 '  ROUTINE 'MODIFIER'
950 '
960 GOSUB 1070
970 PRINT CHR$(7)
980 CLS:LOCATE 10,2:PRINT"ENTREZ VOS VALEURS."
990 FOR I=1 TO 8:LOCATE #2,6,I+1:PRINT#2,"    ":LOCATE #2,6,I+1:INPUT #2,
"",VLIG(I):IF VLIG(I)<0 OR VLIG(I)>255 THEN I=I-1
1000 NEXT I
1010 GOSUB 1190
1020 FOR J=1 TO 8:LIG$=BIN$(VLIG(J)):FOR I=1 TO LEN(LIG$):PT=VAL(MID$(LIG
$,I,1)):CAR(8-LEN(LIG$)+I,J)=PT:NEXT I,J
1030 FLAG=1:GOSUB 850:FLAG=0
1040 LOCATE 2,2:PRINT CHR$(137);" REDEF'CAR ";CHR$(138);"  ";CHR$(164)+"1
985 St";CHR$(139);"phane CLOIREC":LOCATE 4,3:PRINT STRING$(9,134)
1050 GOSUB 1190
1060 RETURN
1070 '
1080 '  ROUTINE 'EFFACE'
1090 '
1100 ERASE VLIG,CAR
1110 FOR I=1 TO 8:LOCATE #1,2,I+1:PRINT#1,STRING$(8,129):NEXT I
1120 LOCATE #3,1,7:PRINT#3,SPACE$(48)
1130 PLOT 398,266,1:DRAWR 0,32:DRAWR 66,0:DRAWR 0,-32:DRAWR -66,0
1140 PLOT 528,266,1:DRAWR 0,32:DRAWR 64,0:DRAWR 0,-32:DRAWR -64,0
1150 PLOT 414,274,3:DRAWR 34,0:DRAWR 0,18:DRAWR -34,0:DRAWR 0,-18
1160 PLOT 552,274:DRAWR 18,0:DRAWR 0,18:DRAWR -18,0:DRAWR 0,-18
1170 IF FL1=1 THEN RETURN
1180 DIM CAR(8,8),VLIG(8)
```

```
1190 XMIN=2:XMAX=9:YMIN=2:YMAX=9:A=2:B=2:X=2:Y=2:LOCATE#1,X,Y:IF CAR(X-1,
Y-1)=1 THEN PRINT#1,CHR$(131) ELSE PRINT #1,CHR$(130)
1200 FOR I=1 TO 8:LOCATE #2,2,1+I:PRINT #2,CHR$(136);MID$(STR$(I),2);CHR$
(135);SPC(4-LEN(STR$(VLIG(I))));STR$(VLIG(I)):NEXT
1210 RETURN
1220 '
1230 '  ROUTINE 'MIROIR'
1240 '
1250 FL1=1:CLS:LOCATE 12,2:PRINT"FONCTION EN COURS":PRINT CHR$(7)
1260 GOSUB 1110:ERASE VLIG
1270 DIM CAR1(8,8),VLIG(8)
1280 FOR J=1 TO 8:FOR I=1 TO 8:CAR1(I,J)=CAR(9-I,J):NEXT I,J
1290 FOR J=1 TO 8:FOR I=1 TO 8:CAR(I,J)=CAR1(I,J):NEXT I,J
1300 ERASE CAR1
1310 GOSUB 850
1320 FOR J=1 TO 8:FOR I=1 TO 8:IF CAR(I,J)=1 THEN VLIG(J)=VLIG(J)+2^(8-I)
1330 NEXT I,J
1340 GOSUB 1190
1350 LOCATE 2,2:PRINT CHR$(137);" REDEF'CAR ";CHR$(138);"  ";CHR$(164)+"1
985 St";CHR$(139);"phane CLOIREC":LOCATE 4,3:PRINT STRING$(9,134)
1360 FL1=0
1370 RETURN
1380 '
1390 '  ROUTINE 'SYMBOL'
1400 '
1410 CLS:LOCATE 12,2:PRINT"SYMBOL EN COURS"+CHR$(7)
1420 I=1:IF ADR=&7000 THEN LENDAT=32
1430 POKE ADR,33
1440 POKE ADR+1,0
1450 POKE ADR+2,NUMLIG
1460 POKE ADR+3,0
1470 POKE ADR+4,207
1480 POKE ADR+5,32
1490 POKE ADR+6,25
1500 POKE ADR+7,NUMSYMB
1510 FOR BC=ADR+8 TO ADR+29 STEP 3
1520 POKE BC,44:POKE BC+1,25:POKE BC+2,VLIG(I)
1530 I=I+1
1540 NEXT BC
1550 POKE ADR+32,0
1560 ADR=ADR+33
1570 NUMLIG=NUMLIG+1
```

```
1580 NUMSYMB=NUMSYMB+1
1590 LENDAT=LENDAT+33
1600 FOR att=1 TO 1000:NEXT att
1610 LOCATE 11,2:PRINT"EFFACEMENT  (O/N) ?":IF INKEY(34)=0 THEN GOSUB 110
0 ELSE IF INKEY(46)<>0 THEN 1610
1620 LOCATE 2,2:PRINT CHR$(137);" REDEF'CAR ";CHR$(138);"  ";CHR$(164)+"1
985 St";CHR$(139);"phane CLOIREC":LOCATE 4,3:PRINT STRING$(9,134)
1630 RETURN
1640 '
1650 '  IMPLANTATION DU L.M.
1660 '
1670 RESTORE 1770
1680 FOR BC=&8000 TO &8000+11
1690 READ C$:C=VAL("&"+C$)
1700 POKE BC,C
1710 NEXT
1720 P$=HEX$(LENDAT)
1730 IF LEN(P$)<>4 THEN P$="0"+P$:GOTO 1730
1740 POKE &8007,VAL("&"+RIGHT$(P$,2))
1750 POKE &8008,VAL("&"+LEFT$(P$,2))
1760 RETURN
1770 DATA 21,00,70,11,70,01
1780 DATA 01,00,00,ED,B0,C9
1790 '
1800 '  INSTRUCTIONS
1810 '
1820 MODE 2:INK 1,0
1830 LOCATE 16,1:PRINT"REDEF'CAR":LOCATE 16,2:PRINT STRING$(9,133)
1840 PRINT:PRINT:PRINT:PRINT"   Redef'car est, comme son nom l'indique, u
n redefinisseur de caracteres. Il   vous permettra de creer facilement to
us les types de symboles graphiques dont   vous pourrez avoir besoin dans
 vos propre programmes."
1850 PRINT:PRINT:PRINT"   Voici les commandes que cet utilitaire met a vo
tre disposition:"
1860 PRINT:PRINT SPC(10);"- EFFACE  : vide le caractere qui etait a l'aff
ichage"
1870 PRINT SPC(10);"- INVERSE : produit une image inversee du caractere d
e travail"
1880 PRINT SPC(10);"- MODIFIE : permet de taper les codes d'un symbole po
ur modification"
1890 PRINT SPC(10);"- MIROIR  : retourne le caractere de travail"
1900 PRINT SPC(10);"- SYMBOL  : stocke le caractere de travail en memoire
```

```
"
1910 PRINT SPC(10);"- QUITTE  : arrete le programme et affiche les caract
eres avant ete":PRINT SPC(22);"stockes avec la fonction 'SYMBOL'"
1920 LOCATE 1,25:PRINT"Appuvez sur une touche pour retour au programme...
"
1930 CALL &BB18:CALL &BB18
1940 RETURN
```

TRUCS ET ASTUCES

Gérard MATTERN — 92 VANVES

Sous l'article MIRAGES de Denis BOURQUIN, j'ai cru voir, dans le premier programme, un ''déplombeur''... ou je ne m'y connais guère. Ce cher Denis s'est cependant beaucoup fatigué, car on peut faire la même chose en plus court, ...même beaucoup plus court. Voici le procédé : Après initialisation de la machine (SHIFT CTRL ESC), tapez les POKE suivants dans l'ordre :

POKE &AC03,&AE:
POKE &AC02,&45:
POKE &AC01,&32...

et c'est tout !
Après un Bad, le programme est toujours en mémoire, et on peut le lister, le modifier et le sauvegarder, protégé ou non.
Principe de fonctionnement : Après un LOAD, et au retour à READY, l'AMSTRAD vérifie si le programme qui vient d'être lu était protégé ou non en testant l'octet &AE45 et détruit tout s'il y avait protection. L'astuce consiste à intercepter le READY et à remettre à zéro l'octet en AE45 avant le test. Ceci se fait par le renvoi en AC01.
Voilà, peut-être pourrez-vous ouvrir avec ça la rubrique ''Trucs et Astuces''. En tout cas, voici un nouveau sujet de concours pour lequel je pense être bien placé : le meilleur utilitaire en une ligne (demi-ligne !) de Basic.
Je termine par une question que je n'ai pas résolue : Si dans une routine binaire on veut lire un fichier par CIOPEN (BC77H), on lit bien sur cassette quand on fonctionne sur cassette (ITAPE), mais sur disque, il se passe des choses curieuses :
— Si on exécute la routine par RUN (chargment disque et exécution), ...on lit la cassette !
— Si on charge par LOAD, puis exécute par CALL (à l'adresse de départ de RUN), ...on lit le disque !
Vos lecteurs experts, auraient-ils une idée sur la question (bug ?).

ATTENTION ! Ces 3 POKE ne sont valables que pour le CPC 464.

4

CARRÉ MAGIQUE

Vannary TEA

Vous aimez jouer avec les nombres ? Voici un programme capable de générer des carrés magiques. Comment ? Vous ne connaissez pas ? Essayez pour voir !

Un carré magique est un carré divisé en cellules dans lesquelles les nombres entiers à partir de 1 sont arrangés de telle manière que les sommes de chaque ligne, de chaque colonne et de chaque diagonale soient égales.
Ce programme vous calcule tous les carrés magiques d'ordre N impair dont le maximum est fixé à 91. En effet, à partir de 93, la mémoire déborde (Memory Full). Le tableau B (K,L) est un tableau intermédiaire utilisé pour le remplissage des cases par des nombres T qui sont transférés au fur et à mesure dans le tableau A (I, J) après vérification que ladite case est bien "inoccupée" ; dans le cas contraire, il faut le mettre ailleurs (voir organigramme).
L'impression du carré magique se fait en alignement par lignes et colonnes uniquement jusqu'à la dimension 14 ; au delà, comme il est impossible d'imprimer toutes les colonnes sur une même ligne, il est donc affiché ligne par ligne. D'autre part, il y a possibilité de vérifier la fiabilité du carré magique affiché.
La durée de calcul d'un tableau 91 x 91 est d'environ 5 minutes.

```
130 '
140 CLS : MODE 0
150 LOCATE 3,10:PRINT STRING$(16,CHR$(210))
160 LOCATE 3,11:PRINT STRING$(1,CHR$(211))
170 LOCATE 18,11:PRINT STRING$(1,CHR$(209))
180 LOCATE 3,12:PRINT STRING$(1,CHR$(211))
190 LOCATE 18,12:PRINT STRING$(1,CHR$(209))
200 LOCATE 3,13:PRINT STRING$(1,CHR$(211))
210 LOCATE 18,13:PRINT STRING$(1,CHR$(209))
```

```
220 LOCATE 3,14:PRINT STRING$(1,CHR$(211))
230 LOCATE 18,14:PRINT STRING$(1,CHR$(209))
240 LOCATE 3,15:PRINT STRING$(1,CHR$(211))
250 LOCATE 18,15:PRINT STRING$(1,CHR$(209))
260 LOCATE 3,16:PRINT STRING$(16,CHR$(208))
270 LOCATE 4,13:PRINT CHR$(67)+CHR$(65)+CHR$(82)+CHR$(82)+CHR$(69)+"  "+C
HR$(77)+CHR$(65)+CHR$(71)+CHR$(73)+CHR$(81)+CHR$(85)+CHR$(69)
280 FOR F=1 TO 1200 : NEXT F
290 '
300 '
310 CLS : MODE 2
320 DEFINT A-M,O-Z
330 PRINT:PRINT:PRINT:PRINT
340 PRINT" ---> ENTREZ LA DIMENSION DU CARRE MAGIQUE :";:INPUT N
350 PRINT
360 IF N<>INT(N) THEN PRINT CHR$(7):PRINT" * LA DIMENSION DU CARRE MAGIQU
E DOIT ETRE UN NOMBRE ENTIER NATUREL. DESOLE !":GOSUB 1740:GOTO 310
370 IF N<3 THEN PRINT CHR$(7):PRINT" * LE CARRE MAGIQUE NE PEUT EXISTER Q
UE POUR UNE DIMENSION SUPERIEURE A 2":GOSUB 1740:GOTO 310
380 IF N>91 THEN PRINT CHR$(7):PRINT" * IL EST IMPOSSIBLE DE FAIRE LE CAL
CUL POUR UN CARRE MAGIQUE D'ORDRE                    SUPERIEUR A  91, CAR LA
 MEMOIRE DEBORDE. VRAIMENT DESOLE !":GOSUB 1740:GOTO 310
390 Q=INT(N/2)
400 IF N/2=Q AND N<=91 THEN PRINT CHR$(7):PRINT" * JE NE TRAITE QUE DES C
ARRES MAGIQUES D'ORDRE IMPAIR. DESOLE !":GOSUB 1740:GOTO 310
410 M=N+1
420 DIM A(N,N),B(M,M)
430 '
440 ' ---------------------INITIALISATION DU TABLEAU A -------------------
-----
450 '
460 FOR I=1 TO N
470 FOR J=1 TO N
480 A(I,J)=0
490 NEXT J : NEXT I
500 '
510 ' ---------------------INITIALISATION DU TABLEAU B -------------------
----
520 '
530 FOR K=1 TO M
540 FOR L=1 TO M
550 B(K,L)=0
```

```
560 NEXT L: NEXT K
570 '
580 ' -------------------- CALCUL DU CARRE MAGIQUE --------------------
---
590 '
600 PRINT:PRINT
610 FOR F=1 TO N:NEXT F : PRINT" >>> PATIENCE S.V.P., JE CALCULE <<< "
620 PRINT:PRINT
630 T=0:I=0:J=0:K=0:L=0
640 K=CINT(N/2)
650 L=INT(N/2)
660 K=K+1
670 L=L+1
680 T=T+1
690 B(K,L)=T
700 IF K>N THEN GOTO 740
710 IF L>N THEN GOTO 730
720 I=K : J=L : GOTO 770
730 I=K : J=M-N : GOTO 770
740 IF L>N THEN GOTO 760
750 I=M-N : J=L : GOTO 770
760 I=M-N : J=N
770 IF A(I,J)=0 THEN GOTO 820
780 I=I+1
790 IF I>N THEN I=M-N
800 IF J>=2 AND J<N OR L=N THEN J=J-1
810 GOTO 770
820 A(I,J)=B(K,L)
830 CPT=CPT+1
840 IF CPT<>N^2 THEN K=I : L=J : GOTO 660
850 PRINT:PRINT
860 IF N>13 THEN GOTO 1060
870 '
880 ' ----------- IMPRESSION DU CARRE MAGIQUE D'ORDRE INFERIEUR A 15-----
-------
890 '
900 CLS : MODE 2
910 PRINT TAB(18)"******** CARRE MAGIQUE D'ORDRE :";N;"********"
920 PRINT:PRINT:PRINT
930 D=0
940 FOR I=1 TO N
950 FOR J=1 TO N
```

```
960 D=D+1
970 PRINT TAB(3*D) A(I,J) ;
980 D=D+1
990 NEXT J
1000 D=0
1010 NEXT I
1020 GOTO 1180
1030 '
1040 ' ------ IMPRESSION DU CARRE MAGIQUE D'ORDRE SUPERIEUR OU EGAL A 15
------
1050 '
1060 PRINT CHR$(7): PRINT"POUR UN CARRE MAGIQUE D'ORDRE SUPERIEUR A 13, I
L EST IMPOSSIBLE D'AFFICHER      TOUTES LES COLONNES DU CARRE MAGIQUE SUR
 UNE MEME LIGNE. AUSSI, ON SE PROPOSE DE L'AFFICHER LIGNE PAR LIGNE.":GOS
UB 1740
1070 PRINT:PRINT:PRINT
1080 CLS : MODE 2
1090 PRINT TAB(18) "******** CARRE MAGIQUE D'ORDRE : ";N;"********"
1100 PRINT:PRINT:PRINT
1110 FOR I=1 TO N
1120 PRINT"LIGNE ";I;": ";
1130 FOR J=1 TO N
1140 PRINT A(I,J);
1150 NEXT J
1160 PRINT:PRINT:PRINT
1170 NEXT I
1180 PRINT:PRINT:PRINT:PRINT" ---> VOULEZ-VOUS EFFECTUER UNE VERIFICATION
 DE CE CARRE MAGIQUE (O/N)  ";:INPUT R$
1190 IF R$<>"O" AND R$<>"o" AND R$<>"N" AND R$<>"n" THEN PRINT CHR$(7):PR
INT:PRINT"ERREUR !!! REPONDEZ PAR O OU N":GOTO 1180
1200 IF R$="N" OR R$="n" THEN GOSUB 1670 : GOTO 1610
1210 PRINT:PRINT
1220 CLS : MODE 2
1230 PRINT TAB(10) "---- LA SOMME DE CHAQUE LIGNE DU CARRE MAGIQUE ----"
1240 PRINT:PRINT:PRINT
1250 FOR I=1 TO N
1260 FOR J=1 TO N
1270 S!=S!+A(I,J)
1280 NEXT J
1290 PRINT:PRINT TAB(16)"LIGNE ";I;":   SOMME =";S!
1300 S!=0
1310 NEXT I
```

```
1320 PRINT:PRINT:PRINT:PRINT
1330 GOSUB 2040
1340 CLS : MODE 2
1350 PRINT TAB(10) "---- LA SOMME DE CHAQUE COLONNE DU CARRE MAGIQUE ----
"
1360 PRINT:PRINT:PRINT
1370 FOR J=1 TO N
1380 FOR I=1 TO N
1390 S!=S!+A(I,J)
1400 NEXT I
1410 PRINT:PRINT TAB(16)"COLONNE ";J;":   SOMME =";S!
1420 S!=0
1430 NEXT J
1440 PRINT:PRINT:PRINT
1450 GOSUB 2040
1460 CLS : MODE 2
1470 PRINT"--- LA SOMME DE CHACUNE DES 2 DIAGONALES PRINCIPALES DU CARRE
MAGIQUE ---"
1480 PRINT:PRINT:PRINT
1490 FOR I=1 TO N
1500 J=I
1510 S!=S!+A(I,J)
1520 NEXT I
1530 PRINT:PRINT TAB(10)"SOMME DE LA 1ERE DIAGONALE :";S!
1540 S!=0
1550 FOR I=N TO 1 STEP -1
1560 J=I
1570 S!=S!+A(I,J)
1580 NEXT I
1590 PRINT:PRINT TAB(10)"SOMME DE LA 2EME DIAGONALE :";S!
1600 GOSUB 1670
1610 PRINT:PRINT:PRINT" ===> VOULEZ-VOUS FAIRE UN AUTRE CARRE MAGIQUE (O/
N) ";:INPUT A$
1620 IF A$<>"O" AND A$<>"o" AND A$<>"N" AND A$<>"n" THEN PRINT CHR$(7):PR
INT:PRINT"ERREUR !!!.REENTREZ O OU N": GOTO 1610
1630 IF A$="O" OR A$="o" THEN RUN
1640 PRINT:PRINT
1650 END
1660 PRINT:PRINT
1670   ----------  SOUS-PROGRAMME DE REINITIALISATION DES VARIABLES -----
------
1680
```

```
1690 ERASE A,B
1700 RETURN
1710 '
1720 '   ---------------- SOUS-PROGRAMME DE TEST DE LA TOUCHE -----------
------
1730 '
1740 SYMBOL AFTER 30
1750 SYMBOL 128,63,63,33,33,1,1,1,1
1760 SYMBOL 129,252,252,132,132,128,128,128,128
1770 SYMBOL 130,1,1,1,1,3,3
1780 SYMBOL 131,128,128,128,128,192,192
1790 SYMBOL 132,15,15,112,112,224,224,224,224
1800 SYMBOL 133,224,224,56,56,28,28,28,28
1810 SYMBOL 134,224,224,112,112,15,15
1820 SYMBOL 135,28,28,56,56,224,224
1830 SYMBOL 136,56,56,56,56,56,56,56,56
1840 SYMBOL 137,28,28,28,28,28,28,28,28
1850 SYMBOL 138,56,56,56,56,15,15
1860 SYMBOL 139,28,28,28,28,240,240
1870 SYMBOL 140,31,31,112,112,224,224,224,224
1880 SYMBOL 141,240,240,28,28
1890 SYMBOL 142,224,224,112,112,31,31
1900 SYMBOL 143,0,0,28,28,240,240
1910 SYMBOL 144,56,56,56,56,56,56,63,63
1920 SYMBOL 145,28,28,28,28,28,28,252,252
1930 SYMBOL 146,56,56,56,56,56,56
1940 SYMBOL 147,28,28,28,28,28,28
1950 SYMBOL 148,255,255,56,56,56,57,63,63
1960 SYMBOL 149,252,252,12,12,0,128,128,128
1970 SYMBOL 150,57,56,56,56,255,255
1980 SYMBOL 151,128,0,12,12,252,252
1990 SYMBOL 152,1,3,6,12,24,48,96,192
2000 SYMBOL 153,192,96,48,24,12,6,3,1
2010 SYMBOL 154,0,0,0,0,0,0,0,0
2020 SYMBOL 156,128,192,96,48,24,12,6,3
2030 SYMBOL 157,3,6,12,24,48,96,192,128
2040 PRINT:PRINT
2050 LOCATE 33,23:PRINT CHR$(152);:PRINT CHR$(154);:PRINT CHR$(128);:PRIN
T CHR$(129);:PRINT CHR$(132);:PRINT CHR$(133);:PRINT CHR$(136);:PRINT CHR
$(137);:PRINT CHR$(140);:PRINT CHR$(141);: PRINT CHR$(144);:PRINT CHR$(14
5);:PRINT CHR$(148);:PRINT CHR$(149)
2060 LOCATE 47,23:PRINT CHR$(154);:PRINT CHR$(156);
```

```
2070 LOCATE 33,24:PRINT CHR$(153);:PRINT CHR$(154);:PRINT CHR$(130);:PRIN
T CHR$(131);:PRINT CHR$(134);:PRINT CHR$(135);:PRINT CHR$(138);:PRINT CHR
$(139);:PRINT CHR$(142);:PRINT CHR$(143);:PRINT CHR$(146);:PRINT CHR$(147
);:PRINT CHR$(150);:PRINT CHR$(151);
2080 LOCATE 47,24:PRINT CHR$(154);:PRINT CHR$(157);
2090 IF INKEY$="" THEN GOTO 2090
2100 RETURN
```

4

MAZE

G. MANKOWSKI

Vous êtes enfermé dans un labyrinthe de 10 étages, dont vous devez sortir au plus vite. La porte de sortie est fermée par 3 cadenas commandés par des clés rangées dans des coffres répartis sur chacun des dix étages.

Vous avez deviné ? Votre travail consiste à retrouver les 3 clés, mais chaque coffre n'en contient pas forcément une ! Pour ouvrir les coffres, il faut d'abord trouver la combinaison de 5 lettres. Pour corser le tout, vous êtes poursuivi par des fantômes malicieux, qui vous subtilisent les clés pour les remettre à leur place, dans les coffres. Le reste de la règle du jeu est énoncé par le programme. A vos claviers !

```
10 '************************
20 '*         MAZE         *
30 '*         pour         *
40 '*   AMSTRAD CPC 464    *
50 '*          par         *
60 '*      G.MANKOWSKI     *
70 '*        mai 1985      *
80 '************************
90 MODE 1
100 DEFINT A-Z
110 CLS:LOCATE 1,10:PRINT "Jouez-vous sur un ecran monochrome ou  couleu
r (M/C) ?"
120 R$=UPPER$(INKEY$):IF R$="M" THEN 140 ELSE IF R$<>"C" THEN 120
130 IF R$="C" THEN 150
140 BORDER 1:INK 0,1:INK 1,24:INK 2,13:INK 3,8
150 np=1
```

```
160 '****************************
170 '*** caracteres redefinis ***
180 '****************************
190 SYMBOL AFTER 220
200 SYMBOL 244,28,28,9,62,92,28,20,20
210 SYMBOL 245,24,24,255,126,60,60,36,36
220 SYMBOL 246,126,129,189,165,165,189,129,126
230 SYMBOL 247,31,17,31,4,4,60,12,60
240 SYMBOL 248,0,255,84,84,84,84,255,0
250 SYMBOL 249,255,195,195,195,195,195,195,255
260 SYMBOL 250,56,68,148,164,169,146,68,56
270 SYMBOL 251,28,34,73,149,37,41,34,28
280 SYMBOL 252,56,68,146,169,164,148,68,56
290 SYMBOL 253,28,34,41,37,149,73,34,28
300 SYMBOL 221,255,0,255,0,255,0,255,0
310 SYMBOL 222,255,0,63,0,15,0,3,0
320 SYMBOL 223,0,0,3,0,15,0,63,0
330 SYMBOL 224,255,0,252,0,240,0,192,0
340 SYMBOL 225,0,0,192,0,240,0,252,0
350 SYMBOL 226,0,1,3,1,1,1,0,0
360 SYMBOL 227,0,119,85,117,85,119,0,0
370 SYMBOL 228,0,119,69,117,21,119,0,0
380 SYMBOL 229,0,119,21,117,69,119,0,0
390 SYMBOL 230,0,119,85,117,21,119,0,0
400 SYMBOL 231,0,119,69,117,85,119,0,0
410 SYMBOL 232,0,119,21,117,21,119,0,0
420 SYMBOL 233,255,255,192,255,255,3,255,255
430 GOSUB 4310:REM *** instructions ***
440 CLS:PEN 1
450 LOCATE 1,10:PRINT "Le niveau de difficulte choisi(1,2 ou 3)"
460 LOCATE 1,12:PRINT "determine le nombre de coffres(5,4 ou 3)"
470 LOCATE 1,14:PRINT "      qui contiendront une clef"
480 LOCATE 1,18:INPUT "        Quel est votre choix";nidi
490 IF nidi=1 OR nidi=2 OR nidi=3 THEN 500 ELSE 480
500 GOSUB 5390
510 ORIGIN 0,0:CLS
520 WINDOW #1,1,25,1,25
530 WINDOW #2,27,27,15,25
540 WINDOW #3,27,40,4,7
550 WINDOW #4,27,40,1,3
560 WINDOW #5,27,40,9,14
570 WINDOW #7,29,39,15,24
```

```
580 DIM L$(5):DIM FX(5):DIM FY(5)
590 DIM co!(73):DIM si!(73)
600 DEG:FOR i=0 TO 72
610 co!(i)=COS(5*i)
620 si!(i)=SIN(5*i)
630 NEXT i
640 INK 0,0:BORDER 0:PAPER 0
650 IF R$="M" THEN 670
660 INK 3,3:INK 1,9:GOTO 680
670 INK 3,9:INK 1,24
680 PAPER #4,3:CLS #4:PEN 1:LOCATE #4,2,2:PRINT #4,CHR$(250);CHR$(251);"M
ESSAGES";CHR$(252);CHR$(253)
690 PAPER #3,3:CLS #3:PRINT#3:PRINT #3,"  VEUILLEZ       PATIENTER"
700 xx=2:yy=2:xxp=2:yyp=2:fois=0:etsor=INT(10*RND+1)
710 RANDOMIZE TIME:n=INT(10*RND+1)
720 '***************************
730 '** generation labyrinthes **
740 '***************************
750 DIM R(5,6,6):DIM S(2,7,7):RANDOMIZE TIME
760 FOR m=1 TO 5
770 FOR i=0 TO 5
780 FOR j=0 TO 5
790 R(m,i,j)=INT(10*RND+1)
800 NEXT j:NEXT i:NEXT m
810 FOR m=1 TO 2
820 FOR i=0 TO 6
830 FOR j=0 TO 6
840 S(m,i,j)=INT(10*RND+1)
850 IF S(m,0,j)<5 THEN 840
860 NEXT j:NEXT i:NEXT m
870 '***************************
880 '** coordonnees escaliers **
890 '***************************
900 DIM EX(15):DIM EY(15):RANDOMIZE TIME
910 FOR i=1 TO 15
920 EX(i)=2*(INT(10*RND+2)):EY(i)=2*(INT(10*RND+2))
930 FOR j=0 TO i-1:IF ABS(EX(i)-EX(j))<3 AND ABS(EY(i)-EY(j))<3 THEN 920
940 NEXT j
950 NEXT i
960 '**************************
970 '*** coordonnees coffre ***
980 '**************************
```

```
990 DIM cox(10):DIM coy(10):RANDOMIZE TIME
1000 FOR i=1 TO 10
1010 cox(i)=2*(INT(12*RND+1)):coy(i)=2*(INT(12*RND+1))
1020 IF cox(i)=4 AND coy(i)=2 THEN 1010
1030 FOR j=1 TO 15
1040 IF coy(i)<>EY(j) THEN 1060
1050 IF cox(i)=EX(j) OR cox(i)=EX(j)+2 THEN 1010
1060 NEXT j:NEXT i
1070 GOSUB 3720:REM *** coordonnees lettre code ***
1080 DIM cle(10):FOR i=1 TO 6-nidi
1090 j=INT(10*RND+1)
1100 IF cle(j)=0 THEN cle(j)=1 ELSE 1090
1110 NEXT
1120 CLS #3
1130 PAPER #5,0:CLS #5:PEN 1:FOR i=1 TO 5:LOCATE #5,1,i:PRINT #5,CHR$(221
):LOCATE #5,14,i:PRINT #5,CHR$(221):NEXT
1140 LOCATE #5,2,1:PRINT #5,"RECORD:":LOCATE #5,2,3:PRINT #5,"TEMPS:":LOC
ATE #5,2,5:PRINT #5,"CLEFS:"
1150 LOCATE #5,9,1:IF record=0 THEN PRINT#5,"-----" ELSE PRINT#5,USING"##
###":record
1160 '**********************
1170 '*** choix couleurs ***
1180 '**********************
1190 CLS#3:CLS #1:IF R$="M" THEN 1370
1200 cn=n MOD 4
1210 ck=-13*(cn=0)-9*(cn=1)-10*(cn=2)-12*(cn=3)
1220 BORDER ck:INK 0,ck
1230 ON n GOTO 1240,1250,1260,1270,1280,1290,1300,1310,1320,1330
1240 INK 1,0:INK 2,4:INK 3,13:GOTO 1370
1250 INK 1,0:INK 2,4:INK 3,13:GOTO 1370
1260 INK 1,0:INK 2,4:INK 3,15:GOTO 1370
1270 INK 1,0:INK 2,4:INK 3,16:GOTO 1370
1280 INK 1,3:INK 2,1:INK 3,0:GOTO 1370
1290 INK 1,3:INK 2,0:INK 3,12:GOTO 1370
1300 INK 1,3:INK 2,0:INK 3,9:GOTO 1370
1310 INK 1,1:INK 2,4:INK 3,11:GOTO 1370
1320 INK 1,1:INK 2,4:INK 3,13:GOTO 1370
1330 INK 1,1:INK 2,12:INK 3,4:GOTO 1370
1340 '*******************
1350 '** choix du code **
1360 '*******************
```

```
1370 RANDOMIZE TIME:z=INT(50*RND):RESTORE 6120
1380 FOR i=-4 TO 5*(z-1):READ v:NEXT i
1390 READ u1,u2,u3,u4,u5
1400 code$=CHR$(u1)+CHR$(u2)+CHR$(u3)+CHR$(u4)+CHR$(u5)
1410 code$=LOWER$(code$)
1420 FOR L=1 TO 5:L$(l)=MID$(code$,L,1):NEXT L
1430 xx=xx+2
1440 '*** indicateur d'etage ***
1450 FOR b=1 TO 10:LOCATE #2,1,b:PRINT #2,CHR$(249):NEXT b
1460 LOCATE #2,1,11-np:PRINT #2,CHR$(249)
1470 LOCATE #2,1,11-n:PRINT #2,CHR$(143)
1480 np=n
1490 '***************
1500 '*** horloge ***
1510 '***************
1520 DI:PEN 3:PAPER #7,2:CLS #7:LOCATE #7,5,1:PRINT #7,CHR$(226);CHR$(227
):LOCATE #7,1,3:PRINT #7,CHR$(226):CHR$(228):LOCATE #7,10,3:PRINT #7,CHR$
(232)
1530 LOCATE #7,1,8:PRINT #7,CHR$(226);CHR$(229):LOCATE #7,10,8:PRINT #7,C
HR$(231):LOCATE #7,6,10:PRINT #7,CHR$(230)
1540 ORIGIN 536,96
1550 MOVE 62*co!(0),62*si!(0):FOR i=1 TO 72:DRAW 62*co!(i),62*si!(i):NEXT
 i
1560 MOVE 52*co!(0),52*si!(0):FOR i=1 TO 72:DRAW 52*co!(i),52*si!(i):NEXT
 i
1570 FOR t=30 TO 330 STEP 60:MOVE 52*COS(t),52*SIN(t):DRAW 62*COS(t),62*S
IN(t):NEXT t:ORIGIN 0,0:EI
1580 '***************************
1590 '* construction labyrinthe *
1600 '***************************
1610 CLS #1
1620 DI:FOR i=0 TO 8 STEP 2:MOVE i,0:DRAWR 0,398:MOVE 398-i,0:DRAWR 0,398
:MOVE 0,i:DRAWR 398,0:MOVE 0,398-i:DRAWR 398,0:NEXT:EI
1630 PEN 1:FOR i=3 TO 23 STEP 2
1640 FOR j=3 TO 23 STEP 2
1650 LOCATE i,j:PRINT CHR$(144)
1660 NEXT j:NEXT i
1670 k=-(n=1 OR n=6)-2*(n=2 OR n=7)-3*(n=3 OR n=8)-4*(n=4 OR n=9)-5*(n=5
OR n=10)
1680 FOR i=0 TO 5
1690 FOR j=0 TO 5
```

```
1700 x=16*(22-4*i)+7
1710 y=16*(22-4*j)+7
1720 ON R(k,i,j) GOSUB 2920,2930,2940,2950,2960,2970,2980,2990,3000,3010
1730 NEXT j:NEXT i
1740 k=-(n>0)-(n>5)
1750 FOR i=0 TO 6
1760 FOR j=0 TO 6
1770 x=16*(24-4*i)+7
1780 y=16*(24-4*j)+7
1790 ON S(k,i,j) GOSUB 2920,2930,2940,2950,2960,2970,2980,2990,3000,3010

1800 NEXT j:NEXT i
1810 ON nc GOSUB 3410,3420,3430,3440,3450,3460,3470,3480,3490,3500
1820 GOSUB 3510
1830 PEN 1:LOCATE cox(n),coy(n):PRINT CHR$(246)
1840 PEN 3:FOR j=1 TO 5:LOCATE codex(n,j),codey(n,j):PRINT CHR$(164):NEXT
 j
1850 IF n=etsor THEN fois=fois+1 ELSE 1910
1860 IF fois=1 THEN GOSUB 3970
1870 PEN 1:LOCATE #1,xsor,ysor:PRINT #1,CHR$(233)
1880 '**************************
1890 '** coordonnees fantomes **
1900 '**************************
1910 RANDOMIZE TIME
1920 FX(1)=2*INT(9*RND+4):FY(1)=2
1930 IF FX(1)=cox(n) THEN 1920
1940 FOR j=1 TO 5:IF FX(1)=codex(n,j) THEN 1920 ELSE NEXT j
1950 FX(2)=2:FY(2)=2*INT(10*RND+2)
1960 IF FY(2)=coy(n) THEN 1950
1970 FOR j=1 TO 5:IF FY(2)=codey(n,j) THEN 1950 ELSE NEXT j
1980 FX(3)=24:FY(3)=2*INT(10*RND+2)
1990 IF FY(3)=coy(n) THEN 1980
2000 FOR j=1 TO 5:IF FY(3)=codey(n,j) THEN 1980 ELSE NEXT j
2010 FX(4)=2*INT(12*RND+1):FY(4)=24
2020 IF FX(4)=cox(n) THEN 2010
2030 FOR j=1 TO 5:IF FX(4)=codex(n,j) THEN 2010 ELSE NEXT j
2040 FX(5)=2*INT(12*RND+1):FY(5)=24
2050 IF FX(5)=cox(n) OR FX(5)=FX(4) THEN 2040
2060 FOR j=1 TO 5:IF FX(5)=codex(n,j) THEN 2040 ELSE NEXT j
2070 PEN 2:FOR k=1 TO 5:LOCATE FX(k),FY(k):PRINT CHR$(245):NEXT k
2080 pause=1:EVERY 50,2 GOSUB 2830
2090 EVERY 100,1 GOSUB 2530
```

```
2100 s0=INT(TIME/300):te=0:pause=0:tepeto=0
2110 '************************
2120 DI:PEN 1
2130 LOCATE xxp,yyp:PRINT CHR$(128)
2140 SOUND 1,150,5,5
2150 LOCATE xx,yy:PRINT CHR$(244)
2160 xxp=xx:yyp=yy:EI
2170 xc=16*(xx-1)+7:yc=16*(25-yy)+7
2180 IF pause=1 THEN n=INT(10*RND+1):xx=xx-2:temps=temps+180:LOCATE #5,8,
3:PRINT #5,USING "#####";temps:GOTO 1190
2190 c$=INKEY$:IF c$="" THEN 2190
2200 IF c$="P" OR c$="p" THEN GOSUB 3570
2210 c=ASC(c$)
2220 IF c<240 OR c>243 THEN 2190
2230 cc=c-239
2240 t=TEST(xc+16*(cc=3)-16*(cc=4),yc-16*(cc=1)+16*(cc=2))
2250 IF t=1 THEN 2190
2260 xt=xc-8*(cc=1)-8*(cc=2)+24*(cc=3)-40*(cc=4)
2270 t1=TEST(xt,yc-40*(cc=1)+24*(cc=2)-8*(cc=3)-8*(cc=4))
2280 t2=TEST(xt,yc-38*(cc=1)+26*(cc=2)-6*(cc=3)-6*(cc=4))
2290 t3=TEST(xt,yc-36*(cc=1)+28*(cc=2)-4*(cc=3)-4*(cc=4))
2300 tt=t1+2*t2+3*t3
2310 IF tt=2 THEN 3050
2320 IF tt=5 THEN GOTO 4060
2330 IF tt=18 THEN GOTO 4170
2340 IF tt=6 THEN GOTO 2190
2350 IF tt=3 AND nbcle=3 THEN GOTO 5460
2360 IF tt=3 AND nbcle<3 THEN GOTO 2120
2370 FOR i=0 TO 4 STEP 2
2380 t4=TEST(xc-(2+i)*(cc=1)-(2+i)*(cc=2)+(26+i)*(cc=3)-(32+i)*(cc=4),yc-
36*(cc=1)+28*(cc=2)-4*(cc=3)-4*(cc=4))
2390 IF t4>0 THEN 2410
2400 NEXT i:GOTO 2460
2410 xxs=xx-2*(cc=4)+2*(cc=3)
2420 yys=yy+2*(cc=1)-2*(cc=2)
2430 FOR i=1 TO 5
2440 IF (codex(n,i)=xxs) AND (codey(n,i)=yys) THEN 2480
2450 NEXT i
2460 yy=yy+2*(cc=1)-2*(cc=2)
2470 xx=xx+2*(cc=3)-2*(cc=4):GOTO 2120
2480 PEN 2:LOCATE xxs,yys:PRINT MID$(code$,i,1)
2490 GOTO 2120
```

```
2500 '************************
2510 '** deplacement fantome **
2520 '************************
2530 DI:IF pause=1 OR pause=2 THEN 2790
2540 FOR I=1 TO 5:A=INT(4*RND+1)
2550 xf=16*(FX(I)-1)+7:yf=16*(25-FY(I))+7
2560 PEN 2:ON A GOTO 2570,2610,2650,2690
2570 IF TEST(xf-20,yf)=1 THEN 2780
2580 IF TEST(xf-24,yf+4)=1 AND TEST(xf-24,yf+6)=0 THEN 2730
2590 IF TEST(xf-24,yf+6)+TEST(xf-26,yf+4)+TEST(xf-28,yf+4)+TEST(xf-30,yf+
4))0 THEN 2780
2600 LOCATE FX(I),FY(I) :PRINT CHR$(128):FX(I)=FX(I)-2:SOUND 1,602,5,5:LO
CATE FX(I),FY(I):PRINT CHR$(245):GOTO 2780
2610 IF TEST(xf+20,yf)=1 THEN 2780
2620 IF TEST(xf+40,yf+4)=1 AND TEST(xf+40,yf+6)=0 THEN 2730
2630 IF TEST(xf+40,yf+6)+TEST(xf+32,yf+4)+TEST(xf+34,yf+4)+TEST(xf+36,yf+
4))0 THEN 2780
2640 LOCATE FX(I),FY(I) :PRINT CHR$(128):FX(I)=FX(I)+2:SOUND 1,602,5,5:LO
CATE FX(I),FY(I):PRINT CHR$(245):GOTO 2780
2650 IF TEST(xf,yf+20)=1 THEN 2780
2660 IF TEST(xf+8,yf+36)=1 AND TEST(xf+8,yf+38)=0 THEN 2730
2670 IF TEST(xf+8,yf+38)+TEST(xf+2,yf+36)+TEST(xf+4,yf+36)+TEST(xf+6,yf+3
6))0 THEN 2780
2680 LOCATE FX(I),FY(I) :PRINT CHR$(128):FY(I)=FY(I)-2:SOUND 1,602,5,5:LO
CATE FX(I),FY(I):PRINT CHR$(245):GOTO 2780
2690 IF TEST(xf,yf-20)=1 THEN 2780
2700 IF TEST(xf+8,yf-28)=1 AND TEST(xf+8,yf-26)=0 THEN 2730
2710 IF TEST(xf+8,yf-26)+TEST(xf+2,yf-28)+TEST(xf+4,yf-28)+TEST(xf+6,yf-2
8))0 THEN 2780
2720 LOCATE FX(I),FY(I) :PRINT CHR$(128):FY(I)=FY(I)+2:SOUND 1,602,5,5:LO
CATE FX(I),FY(I):PRINT CHR$(245):GOTO 2780
2730 IF nbcle=0 THEN 2780
2740 LOCATE #5,nbcle+8,5:PRINT #5,CHR$(128):nbcle=nbcle-1
2750 GOSUB 5800:CLS #3:PRINT #3,"1 cle de moins"
2760 ce=INT(10*RND+1)
2770 IF cle(ce)=1 THEN 2760 ELSE cle(ce)=1
2780 NEXT I
2790 EI:RETURN
2800 '************************
2810 '*** mouvement horloge ***
2820 '************************
2830 IF pause=1 THEN 2870
```

```
2840 DEG:te=te+2
2850 IF te<360 THEN 2870
2860 pause=1:te=0
2870 MOVE 536,96:DRAWR 52*SIN(te),52*COS(te)
2880 RETURN
2890 '***************************
2900 '* construction labyrinthe *
2910 '***************************
2920 MOVE x,y:DRAWR 32,0:MOVE x,y+2:DRAWR 32,0:RETURN
2930 MOVE x,y:DRAWR 32,0:MOVE x,y+2:DRAWR 32,0:MOVE x,y:DRAWR -32,0:MOVE
x,y+2:DRAWR -32,0:RETURN
2940 MOVE x,y:DRAWR 32,0:MOVE x,y+2:DRAWR 32,0:MOVE x,y:DRAWR 0,32:MOVE x
+2,y:DRAWR 0,32:RETURN
2950 MOVE x,y:DRAWR 32,0:MOVE x,y+2:DRAWR 32,0:MOVE x,y:DRAWR 0,-32:MOVE
x+2,y:DRAWR 0,-32:RETURN
2960 MOVE x,y:DRAWR -32,0:MOVE x,y+2:DRAWR -32,0:RETURN
2970 MOVE x,y:DRAWR -32,0:MOVE x,y+2:DRAWR -32,0:MOVE x,y:DRAWR 0,32:MOVE
 x+2,y:DRAWR 0,32:RETURN
2980 MOVE x,y:DRAWR -32,0:MOVE x,y+2:DRAWR -32,0:MOVE x,y:DRAWR 0,-32:MOV
E x+2,y:DRAWR 0,-32:RETURN
2990 MOVE x,y:DRAWR 0,32:MOVE x+2,y:DRAWR 0,32:RETURN
3000 MOVE x,y:DRAWR 0,32:MOVE x+2,y:DRAWR 0,32:MOVE x,y:DRAWR 0,-32:MOVE
x+2,y:DRAWR 0,-32:RETURN
3010 MOVE x,y:DRAWR 0,-32:MOVE x+2,y:DRAWR 0,-32:RETURN
3020 '********************
3030 '*** saut d'etage ***
3040 '********************
3050 yy=yy+2*(cc=1)-2*(cc=2)
3060 xx=xx+2*(cc=3)-2*(cc=4)
3070 pause=1:te=0:s1=INT(TIME/300):temps=temps+s1-s0+tepeto:LOCATE #5,8,3
:PRINT #5,USING "#####":temps
3080 ON n GOTO 3090,3120,3150,3180,3210,3240,3270,3300,3330,3360
3090 IF xx=EX(1) AND yy=EY(1) THEN n=2:GOTO 1170
3100 IF xx=EX(2) AND yy=EY(2) THEN n=3:GOTO 1170
3110 IF xx=EX(3) AND yy=EY(3) THEN n=4:GOTO 1170
3120 IF xx=EX(1) AND yy=EY(1) THEN n=1:GOTO 1170
3130 IF xx=EX(4) AND yy=EY(4) THEN n=3:GOTO 1170
3140 IF xx=EX(5) AND yy=EY(5) THEN n=5:GOTO 1170
3150 IF xx=EX(6) AND yy=EY(6) THEN n=4:GOTO 1170
3160 IF xx=EX(2) AND yy=EY(2) THEN n=1:GOTO 1170
3170 IF xx=EX(4) AND yy=EY(4) THEN n=2:GOTO 1170
```

```
3180 IF xx=EX(6) AND yy=EY(6) THEN n=3:GOTO 1170
3190 IF xx=EX(8) AND yy=EY(8) THEN n=10:GOTO 1170
3200 IF xx=EX(3) AND yy=EY(3) THEN n=1:GOTO 1170
3210 IF xx=EX(7) AND yy=EY(7) THEN n=6:GOTO 1170
3220 IF xx=EX(11) AND yy=EY(11) THEN n=8:GOTO 1170
3230 IF xx=EX(5) AND yy=EY(5) THEN n=2:GOTO 1170
3240 IF xx=EX(9) AND yy=EY(9) THEN n=8:GOTO 1170
3250 IF xx=EX(7) AND yy=EY(7) THEN n=5:GOTO 1170
3260 IF xx=EX(13) AND yy=EY(13) THEN n=9:GOTO 1170
3270 IF xx=EX(10) AND yy=EY(10) THEN n=10:GOTO 1170
3280 IF xx=EX(14) AND yy=EY(14) THEN n=8:GOTO 1170
3290 IF xx=EX(15) AND yy=EY(15) THEN n=9:GOTO 1170
3300 IF xx=EX(9) AND yy=EY(9) THEN n=6:GOTO 1170
3310 IF xx=EX(11) AND yy=EY(11) THEN n=5:GOTO 1170
3320 IF xx=EX(14) AND yy=EY(14) THEN n=7:GOTO 1170
3330 IF xx=EX(12) AND yy=EY(12) THEN n=10:GOTO 1170
3340 IF xx=EX(13) AND yy=EY(13) THEN n=6:GOTO 1170
3350 IF xx=EX(15) AND yy=EY(15) THEN n=7:GOTO 1170
3360 IF xx=EX(8) AND yy=EY(8) THEN n=4:GOTO 1170
3370 IF xx=EX(10) AND yy=EY(10) THEN n=7:GOTO 1170
3380 IF xx=EX(12) AND yy=EY(12) THEN n=9:GOTO 1170
3390 GOTO 1170
3400 '************************
3410 P1=EX(1):P2=EY(1):P3=EX(2):P4=EY(2):P5=EX(3):P6=EY(3):RETURN
3420 P1=EX(1):P2=EY(1):P3=EX(4):P4=EY(4):P5=EX(5):P6=EY(5):RETURN
3430 P1=EX(6):P2=EY(6):P3=EX(4):P4=EY(4):P5=EX(2):P6=EY(2):RETURN
3440 P1=EX(6):P2=EY(6):P3=EX(8):P4=EY(8):P5=EX(3):P6=EY(3):RETURN
3450 P1=EX(7):P2=EY(7):P3=EX(11):P4=EY(11):P5=EX(5):P6=EY(5):RETURN
3460 P1=EX(7):P2=EY(7):P3=EX(13):P4=EY(13):P5=EX(9):P6=EY(9):RETURN
3470 P1=EX(10):P2=EY(10):P3=EX(14):P4=EY(14):P5=EX(15):P6=EY(15):RETURN
3480 P1=EX(11):P2=EY(11):P3=EX(14):P4=EY(14):P5=EX(9):P6=EY(9):RETURN
3490 P1=EX(12):P2=EY(12):P3=EX(13):P4=EY(13):P5=EX(15):P6=EY(15):RETURN
3500 P1=EX(12):P2=EY(12):P3=EX(10):P4=EY(10):P5=EX(8):P6=EY(8):RETURN
3510 PEN 1:LOCATE P1,P2:PRINT CHR$(248);CHR$(128):LOCATE P3,P4:PRINT CHR$
(248);CHR$(128):LOCATE P5,P6:PRINT CHR$(248);CHR$(128)
3520 DI:MOVE 16*P1+7,16*(26-P2)+7:DRAWR -32,0:DRAWR 0,-32:DRAWR 32,0:MOVE
 16*P3+7,16*(26-P4)+7:DRAWR -32,0:DRAWR 0,-32:DRAWR 32,0:MOVE 16*P5+7,16*
(26-P6)+7:DRAWR -32,0:DRAWR 0,-32:DRAWR 32,0
3530 MOVE 16*P1+8,16*(26-P2)+8:DRAWR -31,0:DRAWR 0,-32:DRAWR 31,0:MOVE 16
*P3+8,16*(26-P4)+8:DRAWR -31,0:DRAWR 0,-32:DRAWR 31,0:MOVE 16*P5+8,16*(26
-P6)+8:DRAWR -31,0:DRAWR 0,-32:DRAWR 31,0:EI:RETURN
```

```
3540 '***************
3550 '*** percage ***
3560 '***************
3570 GOSUB 5760:CLS#3:PRINT #3," indiquer la   direction de  percage"
3580 c$=INKEY$:IF c$="" THEN 3580
3590 c= ASC(c$):IF c<240 OR c>243 THEN 3580
3600 IF TEST (xc+20*(c=242)-20*(c=243),yc+20*(c=241)-20*(c=240))=1 THEN 3
680
3610 tepe=INT(16*RND+5):te=te+2*tepe:tepeto=tepeto+tepe
3620 GOSUB 5760:CLS#3:PRINT#3,"   duree de      percage:      ";tepe;"sec.
"
3630 ON c-239 GOTO 3640,3650,3660,3670
3640 LOCATE xx,yy-1:PRINT CHR$(128):RETURN
3650 LOCATE xx,yy+1:PRINT CHR$(128):RETURN
3660 LOCATE xx-1,yy:PRINT CHR$(128):RETURN
3670 LOCATE xx+1,yy:PRINT CHR$(128):RETURN
3680 GOSUB 5760:CLS#3:PRINT#3,"    percage     impossible":RETURN
3690 '***************************
3700 '* coordonnees lettre code *
3710 '***************************
3720 DIM codex(10,5):DIM codey(10,5)
3730 FOR i=1 TO 10
3740 FOR j=1 TO 5
3750 codex(i,j)=2*(INT(12*RND+1)):codey(i,j)= 2*(INT(12*RND+1))
3760 IF codex(i,j)=4 AND codey(i,j)=2 THEN 3750
3770 FOR k=0 TO j-1
3780 IF codex(i,j)=codex(i,k) AND codey(i,j)=codey(i,k) THEN 3750
3790 IF codex(i,j)=cox(i) AND codey(i,j)=coy(i) THEN 3750
3800 ON i GOTO 3810,3820,3830,3840,3850,3860,3870,3880,3890,3900
3810 IF codey(i,j)=EY(1) OR codey(i,j)=EY(2) OR codey(i,j)=EY(3) THEN 375
0 ELSE 3910
3820 IF codey(i,j)=EY(1) OR codey(i,j)=EY(4) OR codey(i,j)=EY(5) THEN 375
0 ELSE 3910
3830 IF codey(i,j)=EY(6) OR codey(i,j)=EY(2) OR codey(i,j)=EY(4) THEN 375
0 ELSE 3910
3840 IF codey(i,j)=EY(6) OR codey(i,j)=EY(8) OR codey(i,j)=EY(3) THEN 375
0 ELSE 3910
3850 IF codey(i,j)=EY(7) OR codey(i,j)=EY(11) OR codey(i,j)=EY(5) THEN 37
50 ELSE 3910
3860 IF codey(i,j)=EY(9) OR codey(i,j)=EY(7) OR codey(i,j)=EY(13) THEN 37
50 ELSE 3910
3870 IF codey(i,j)=EY(10) OR codey(i,j)=EY(14) OR codey(i,j)=EY(15) THEN
```

```
3750 ELSE 3910
3880 IF codev(i,j)=EY(9) OR codev(i,j)=EY(14) OR codev(i,j)=EY(11) THEN 3
750 ELSE 3910
3890 IF codev(i,j)=EY(12) OR codev(i,j)=EY(13) OR codev(i,j)=EY(15) THEN
3750 ELSE 3910
3900 IF codev(i,j)=EY(8) OR codev(i,j)=EY(10) OR codev(i,j)=EY(12) THEN 3
750 ELSE 3910
3910 NEXT k
3920 NEXT j
3930 NEXT i:RETURN
3940 '***********************
3950 '*** porte de sortie ***
3960 '***********************
3970 xsor=2*INT(10*RND+2):ysor=2*INT(10*RND+2)
3980 IF xsor=cox(etsor) AND ysor=coy(etsor) THEN 3970
3990 FOR i=1 TO 5
4000 IF xsor=codex(etsor,i) AND ysor=codey(etsor,i) THEN 3970 ELSE NEXT i
4010 IF (xsor=P1 AND ysor=P2) OR (xsor=P1+2 AND ysor=P2) THEN 3970
4020 IF (xsor=P3 AND ysor=P4) OR (xsor=P3+2 AND ysor=P4) THEN 3970
4030 IF (xsor=P5 AND ysor=P6) OR (xsor=P5+2 AND ysor=P6) THEN 3970
4040 RETURN
4050 '************************
4060 '*** ouverture coffre ***
4070 '************************
4080 pause=2:GOSUB 5760:CLS #3:INPUT#3, "Code du coffre ";c$
4090 c$=LOWER$(c$):IF c$=code$ THEN 4110
4100 CLS#3:PRINT #3,"Le code n'est pas bon":pause=0:GOTO 2120
4110 IF cle(n)=1 THEN 4140
4120 GOSUB 5770:CLS #3:PRINT#3,"     BRAVO     mais il n'y a pas de cle."
4130 pause=0:GOTO 2120
4140 PEN 3:LOCATE cox(n),coy(n):PRINT CHR$(247)
4150 GOSUB 5770:CLS #3:PEN 1:PRINT#3,"    BRAVO,    prenez la cle"
4160 '******************
4170 '*** prise clef ***
4180 '******************
4190 c$=INKEY$:IF c$="" THEN 4190
4200 c=ASC(c$):IF c<240 OR c>243 THEN 4190
4210 xcl=xx+2*(c=242)-2*(c=243)
4220 ycl=yy+2*(c=240)-2*(c=241)
4230 IF xcl=cox(n) AND ycl=coy(n) THEN 4240 ELSE 4290
4240 nbcle=nbcle+1:cle(n)=0
4250 IF nbcle=1 THEN LOCATE #5,9,5:PRINT #5,CHR$(247)
```

```
4260 IF nbcle=2 THEN LOCATE #5,10,5:PRINT #5,CHR$(247)
4270 IF nbcle=3 THEN LOCATE #5,11,5:PRINT #5,CHR$(247)
4280 PEN 1:LOCATE cox(n),coy(n):PRINT CHR$(246)
4290 pause=0:GOTO 2120
4300 '********************
4310 '*** instructions ***
4320 '********************
4330 IF R$="C" THEN INK 0,0:INK 1,9:INK 2,11:BORDER 0
4340 PAPER 0:PEN 2:CLS:FOR i=3 TO 37:LOCATE i,1:PRINT CHR$(221):LOCATE i,
9:PRINT CHR$(221):NEXT i
4350 FOR i=2 TO 8:LOCATE 3,i:PRINT CHR$(221):LOCATE 37,i:PRINT CHR$(221):
NEXT i
4360 READ a:IF a=20 THEN 4410
4370 IF a<10 THEN 4380 ELSE 4390
4380 lig=a:col=3:GOTO 4360
4390 col=col+1
4400 LOCATE col,lig:PRINT CHR$(a):GOTO 4360
4410 WINDOW #6,1,40,11,25:PAPER #6,0:CLS #6:PEN 3
4420 FOR i=1 TO 14:PRINT #6:NEXT
4430 GOSUB 5820
4440 PRINT #6,"Voulez-vous les instructions ? (O/N)"
4450 I$=UPPER$(INKEY$):IF I$="N" THEN RETURN ELSE IF I$<>"O" THEN 4450
4460 CLS #6:PRINT #6:PRINT #6:T$="INSTRUCTIONS":GOSUB 5290
4470 T$="Pour faire defiler les instructions,":GOSUB 5280
4480 T$="appuyez sur une touche quelconque.":GOSUB 5290
4490 CALL &BB06
4500 T$="Vous vous trouvez dans un labyrinthe de":GOSUB 5290
4510 T$="10 etages et vous devez en sortir le":GOSUB 5290
4520 T$="plus rapidement possible.":GOSUB 5290
4530 CALL &BB06
4540 T$="Mais, pour pouvoir sortir,il vous":GOSUB 5280
4550 T$="faudra etre en possession des 3 clefs":GOSUB 5290
4560 T$="qui vous permettront de deverrouiller":GOSUB 5290
4570 T$="les 3 cadenas qui condamnent la porte":GOSUB 5290
4580 T$="de sortie.":GOSUB 5290
4590 CALL &BB06
4600 T$="Les clefs sont enfermees dans 5, 4 ou":GOSUB 5280
4610 T$="seulement 3 des 10 coffres que vous":GOSUB 5290
4620 T$="trouverez dans le labyrinthe.":GOSUB 5290
4630 CALL &BB06
4640 T$="L'ouverture d'un coffre necessite la":GOSUB 5280
4650 T$="connaissance d'un code constitue de 5":GOSUB 5290
```

```
4660 T$="lettres formant un nom d'animal.":GOSUB 5290
4670 CALL &BB06
4680 T$="Les lettres du code sont dissimulees":GOSUB 5280
4690 T$="a differents endroits de l'etage. Il":GOSUB 5290
4700 T$="peut arriver qu'une lettre du code ne":GOSUB 5290
4710 T$="puisse pas etre obtenue.De toute facon,":GOSUB 5290
4720 T$="vous n'avez pas besoin de connaitre":GOSUB 5290
4730 T$="toutes les lettres du code pour essayer":GOSUB 5290
4740 T$="d'ouvrir un coffre.":GOSUB 5290
4750 CALL &BB06
4760 T$="Les escaliers vous meneront a un autre":GOSUB 5280
4770 T$="etage, mais pas obligatoirement voisin.":GOSUB 5290
4780 T$="Un escalier relie toujours les deux":GOSUB 5290
4790 T$="memes etages. Un dessin a droite du":GOSUB 5290
4800 T$="labyrinthe vous indiquera l'etage ou":GOSUB 5290
4810 T$="vous vous trouvez.":GOSUB 5290
4820 CALL &BB06
4830 T$="Vous n'avez le droit de rester a un":GOSUB 5280
4840 T$="meme etage qu'un temps limite (180 sec)":GOSUB 5290
4850 T$="Au bout de ce laps de temps, vous serez":GOSUB 5290
4860 T$="transfere automatiquement a un autre":GOSUB 5290
4870 T$="etage. Aussi, surveillez l'horloge.":GOSUB 5290
4880 CALL &BB06
4890 T$="Si vous etes coince, vous pouvez vous":GOSUB 5280
4900 T$="en sortir en utilisant la touche <P>":GOSUB 5290
4910 T$="qui vous permet de percer une cloison.":GOSUB 5290
4920 T$="Mais, attention, suivant la resistance":GOSUB 5290
4930 T$="de la cloison, le temps necessaire au":GOSUB 5290
4940 T$="percage peut etre plus ou moins long:":GOSUB 5290
4950 T$="de 5 a 20 secondes. Aussi, n'utilisez":GOSUB 5290
4960 T$="cette possibilite qu'a bon escient.":GOSUB 5290
4970 CALL &BB06
4980 T$="Tout cela serait tres facile, s'il n'y":GOSUB 5280
4990 T$="avait pas des fantomes qui rodent et":GOSUB 5290
5000 T$="qui ont la mauvaise habitude de vous":GOSUB 5290
5010 T$="derober les clefs et d'aller les":GOSUB 5290
5020 T$="remettre dans un coffre. Bien qu'ils":GOSUB 5290
5030 T$="ne vous poursuivent pas vraiment,":GOSUB 5290
5040 T$="mefiez-vous en cependant":GOSUB 5290
5050 CALL &BB06
5060 T$="Pour vous deplacer utilisez les fleches":GOSUB 5280
5070 PRINT#6:PRINT#6:PRINT#6,"     ";CHR$(244);" : vous"
```

```
5080 PRINT#6:PRINT#6,"      ";CHR$(245);" : les fantomes"
5090 PRINT#6:PRINT#6,"      ";CHR$(246);" : les coffres"
5100 PRINT#6:PRINT#6,"      ";CHR$(247);" : les clefs"
5110 PRINT#6:PRINT#6,"      ";CHR$(248);" : les escaliers"
5120 PRINT#6:PRINT#6,"      ";CHR$(164);" : les lettres du code"
5130 PRINT#6:PRINT#6,"      ";CHR$(233);" : la sortie"
5140 CALL &BB06
5150 T$="REMARQUES":GOSUB 5280
5160 T$="La forme des labyrinthes, les positions":GOSUB 5280
5170 T$="des escaliers, du coffre, des lettres":GOSUB 5290
5180 T$="du code, de la porte de sortie, restent":GOSUB 5290
5190 T$="les memes au cours d'une partie, mais":GOSUB 5290
5200 T$="changent si vous en faites une autre.":GOSUB 5290
5210 CALL &BB06
5220 T$="Lorsque vous arrivez a un nouvel etage,":GOSUB 5280
5230 T$="et meme si vous y etes deja passe, un":GOSUB 5290
5240 T$="un nouveau code est affecte au coffre.":GOSUB 5290
5250 T$="BONNE CHANCE":GOSUB 5280
5260 CALL &BB06
5270 CLS:RETURN
5280 PRINT #6
5290 PRINT #6:PRINT #6
5300 LOCATE #6,INT((40-LEN(T$))/2)+1,25:FOR t=1 TO LEN(T$):PRINT #6,MID$(
T$,t,1);:IF MID$(T$,t,1)<>" " THEN SOUND 1,284,3,5,0,0,1
5310 FOR d=1 TO 50:NEXT d:NEXT t:RETURN
5320 DATA 3,128,221,221,221,221,225,223,221,221,225,128,128,221,221,221,2
21,225,128,128,221,221,221,221,221,221,128,223,221,221,221,221,225,128
5330 DATA 4,128,128,221,221,221,221,221,221,221,221,128,128,128,128,128,2
21,221,128,128,128,128,223,221,221,224,128,221,221,128,128,128,221,128
5340 DATA 5,128,128,221,221,128,221,221,128,221,221,128,223,221,221,221,2
21,221,128,128,128,223,221,221,224,128,128,221,221,221,221,221,224,128
5350 DATA 6,128,128,221,221,128,221,221,128,221,221,128,221,128,128,128,2
21,221,128,128,223,221,221,224,128,128,128,221,221,128,128,128,128,128
5360 DATA 7,128,128,221,221,128,221,221,128,221,221,128,222,221,221,221,2
24,222,221,128,221,221,221,221,221,221,128,222,221,221,221,221,221,128
5370 DATA 9,221,221,221,221,221,221,221,221,71,46,77,65,78,75,79,87,83,75
,73,45,49,57,56,53,20
5380 '*********************
5390 n1$="Victor":h1=5555
5400 n2$="William":h2=6666
5410 n3$="Xavier":h3=7777
```

```
5420 n4$="Yvan":h4=8888
5430 n5$="Zoe":h5=10000
5440 RETURN
5450 '********************
5460 '*** fin de partie ***
5470 '********************
5480 pause=1
5490 CLS#3:CLS#1
5500 s1=INT(TIME/300):temps=temps+s1-s0+tepeto
5510 IF temps>=h5 THEN LOCATE#1,1,10:PRINT#1,"VOTRE SCORE EST TROP":PRINT
#1:PRINT#1,"MAUVAIS POUR POUVOIR":PRINT#1:PRINT#1,"FIGURER AU PALMARES":G
OTO 5580
5520 IF TEMPS<h1 THEN LOCATE#1,1,10:PRINT#1,"TOUTES MES FELICITATIONS:":P
RINT#1:PRINT#1,"VOUS AVEZ BATTU LE RECORD":GOTO 5540
5530 IF TEMPS<h5 THEN PRINT#1,"BRAVO, BELLE PERFORMANCE":PRINT#1:PRINT#1,
"QUI VOUS PERMET DE":PRINT#1:PRINT#1,"FIGURER AU PALMARES":GOTO 5540
5540 INPUT "Quel est votre nom";n$
5550 IF temps<h1 THEN n5$=n4$:n4$=n3$:n3$=n2$:n2$=n1$:n1$=n$ ELSE IF temp
s<h2 THEN n5$=n4$:n4$=n3$:n3$=n2$:n2$=n$ ELSE IF temps<h3 THEN n5$=n4$:n4
$=n3$:n3$=n$ ELSE IF temps<h4 THEN n5$=n4$:n4$=n$ ELSE IF temps<h5 THEN n
5$=n$
5560 tem=temps
5570 IF tem<h1 THEN h5=h4:h4=h3:h3=h2:h2=h1:h1=tem ELSE IF tem<h2 THEN h5
=h4:h4=h3:h3=h2:h2=tem ELSE IF tem<h3 THEN h5=h4:h4=h3:h3=tem ELSE IF tem
<h4 THEN h5=h4:h4=tem ELSE IF tem<h5 THEN h5=tem
5580 CLS#1
5590 FOR i=7 TO 18:LOCATE #1,i,1:PRINT#1,CHR$(238):LOCATE #1,i,7:PRINT#1,
CHR$(238):NEXT
5600 FOR i=2 TO 6:LOCATE #1,7,i:PRINT#1,CHR$(238):LOCATE #1,18,i:PRINT#1,
CHR$(238):NEXT
5610 LOCATE #1,11,3:PRINT#1,"MAZE"
5620 LOCATE #1,9,5:PRINT#1,"PALMARES"
5630 LOCATE #1,3,11:PRINT#1,"1- ";n1$:LOCATE #1,16,11:PRINT#1,USING "####
#";h1;:PRINT#1," sec."
5640 LOCATE #1,3,13:PRINT#1,"2- ";n2$:LOCATE #1,16,13:PRINT#1,USING "####
#";h2;:PRINT#1," sec."
5650 LOCATE #1,3,15:PRINT#1,"3- ";n3$:LOCATE #1,16,15:PRINT#1,USING "####
#";h3;:PRINT#1," sec."
5660 LOCATE #1,3,17:PRINT#1,"4- ";n4$:LOCATE #1,16,17:PRINT#1,USING "####
#";h4;:PRINT#1," sec."
5670 LOCATE #1,3,19:PRINT#1,"5- ";n5$:LOCATE #1,16,19:PRINT#1,USING "####
#";h5;:PRINT#1," sec."
```

```
5680 GOSUB 5820
5690 LOCATE #1,1,22:PRINT#1,"Voulez-vous faire une    autre partie (O/N)
?"
5700 K$=UPPER$(INKEY$):IF K$="N" THEN NEW ELSE IF K$<>"O" THEN 5700
5710 ERASE R,S,EX,EY,cox,coy,cle,codex,codey
5720 temps=0:nbcle=0:record=h1:CLS:GOTO 680
5730 '*****************
5740 '*** BRUITAGES ***
5750 '*****************
5760 SOUND 1,200,15,4:SOUND 1,0,5:SOUND 1,150,15,4:SOUND 1,0,5:SOUND 1,10
0,15,4:SOUND 1,0,5:SOUND 1,80,30,4:RETURN
5770 FOR i=0 TO 700 STEP 100
5780 SOUND 1,2000+2*i,10+i/100,7
5790 NEXT i:RETURN
5800 SOUND 1,284,20,5:SOUND 1,0,2:SOUND 1,284,20,5:SOUND 1,319,20,5:SOUND
 1,358,40,5:RETURN
5810 '*Aupres de ma blonde*
5820 RESTORE 5880
5830 FOR no=1 TO 92
5840 READ fr,du
5850 SOUND 1,fr,du,5
5860 SOUND 2,fr/2,du,3
5870 SOUND 4,fr*2,du,3
5880 DATA 142,20,0,3,142,40,134,20,142,40
5890 DATA 159,20,179,60,0,3,179,20,0,10
5900 DATA 179,20,119,40,0,3,119,20,106,40
5910 DATA 0,3,106,20,119,60,0,3,119,20
5920 DATA 0,10,142,20,0,3,142,40,134,20
5930 DATA 142,40,159,20,179,60,0,3,179,20
5940 DATA 0,10,179,20,119,40,0,3,119,20
5950 DATA 106,40,0,3,106,20,119,60,0,3
5960 DATA 119,20,0,10,119,20,106,40,0,3
5970 DATA 106,20,0,3,106,40,142,20,134,40
5980 DATA 0,3,134,40,0,3,134,20,119,40
5990 DATA 0,3,119,20,0,3,119,40,159,20
6000 DATA 142,60,0,3,142,40,0,10,179,60
6010 DATA 159,20,142,20,134,20,142,60,119,40
6020 DATA 0,10,159,40,134,20,142,40,159,20
6030 DATA 179,40,213,20,239,40,0,10,179,60
6040 DATA 159,20,142,20,134,20,142,60,119,40
6050 DATA 0,10,159,40,134,20,142,40,159,20
6060 DATA 179,60,179,40
```

```
6070 NEXT no
6080 RETURN
6090 '*****************
6100 '***   CODES   ***
6110 '*****************
6120 DATA 65,73,71,76,69
6130 DATA 65,83,80,73,67
6140 DATA 66,73,67,72,69
6150 DATA 66,73,83,79,78
6160 DATA 66,79,69,85,70
6170 DATA 66,85,84,79,82
6180 DATA 67,65,82,80,69
6190 DATA 67,72,73,69,78
6200 DATA 67,82,65,66,69
6210 DATA 67,89,71,78,69
6220 DATA 68,73,78,68,69
6230 DATA 68,73,78,71,79
6240 DATA 68,79,71,85,69
6250 DATA 69,73,68,69,82
6260 DATA 70,85,82,69,84
6270 DATA 71,82,73,86,69
6280 DATA 71,85,69,80,69
6290 DATA 72,69,82,79,78
6300 DATA 72,73,66,79,85
6310 DATA 72,85,80,80,69
6320 DATA 72,89,69,78,69
6330 DATA 73,83,65,82,68
6340 DATA 75,79,65,76,65
6350 DATA 76,65,80,73,78
6360 DATA 76,69,82,79,84
6370 DATA 77,65,78,84,69
6380 DATA 77,69,82,76,69
6390 DATA 77,73,76,65,78
6400 DATA 77,79,82,83,69
6410 DATA 77,79,82,85,69
6420 DATA 77,79,85,76,69
6430 DATA 77,85,76,69,84
6440 DATA 77,85,76,79,84
6450 DATA 79,75,65,80,73
6460 DATA 79,82,86,69,84
6470 DATA 80,65,78,68,65
6480 DATA 80,79,78,69,89
6490 DATA 80,79,85,76,69
6500 DATA 82,65,84,79,78
6510 DATA 83,65,73,71,65
6520 DATA 83,69,82,73,78
6530 DATA 83,73,78,71,69
6540 DATA 84,65,80,73,82
6550 DATA 84,65,84,79,85
6560 DATA 84,65,85,80,69
6570 DATA 84,73,71,82,69
6580 DATA 84,82,85,73,69
6590 DATA 86,65,67,72,69
6600 DATA 86,73,83,79,78
6610 DATA 90,69,66,82,69
```

4

LEXIQUE ANGLAIS/FRANÇAIS

Michel ARCHAMBAULT

Lorsque l'on se débat avec une notice en anglais — c'est hélas fréquent — on se dit souvent "j'ai déjà recherché ce mot dans le dictionnaire et je ne m'en souviens plus". Alors, changeons nos habitudes : avant d'attaquer une lecture anglaise, lançons ce programme fichier.

Mot inconnu ; on le tape au clavier. S'il est déjà au fichier, réponse presque instantanément (environ 1 seconde), s'il ne l'est pas, dictionnaire. Une fois la traduction trouvée, vite option C (= compléter) pour gonfler le fichier de ce nouveau mot anglais et ainsi de suite. A la fin, on sauvegarde le fichier ainsi augmenté. Le mien contient déjà 286 mots (ceux que je risque d'oublier) et ce n'est qu'un début...

Ce logiciel est donc un gestionnaire de fichiers mais "très spécialisé", conçu uniquement pour cet usage, ce qui lui confère des performances que ne pourrait avoir un "gestionnaire à tout faire" du commerce. Y compris certains progiciels tournant sur IBM-PC !

SES PARTICULARITÉS

— Un seul et unique menu. On a supprimé les questions inutiles genre "nom du fichier à charger", puisque c'est toujours le même et qu'il se charge dès le départ...
— Chaque fiche n'a que deux rubriques : mot anglais + sa traduction française. Elle tient sur **une ligne d'écran**, avec son numéro.
— De ce fait, la saisie et la lecture se font en "plein écran". Il est en effet utile de pouvoir lire quelques lignes plus haut la fiche-mot que l'on a demandée il y a quelques minutes... Vous voyez pourquoi.
— Au menu principal, il suffit de taper la lettre initiale de ce que l'on veut faire (sans ENTER). De haut en bas, nous avons :

A : Anglais vers français (c'est la plus utilisée !).
F : Français vers anglais. Eh oui ! La recherche se fait indifféremment dans les deux sens.
C : Compléter le lexique.
M : Modifier une fiche (correction d'erreur).
Viennent ensuite les options moins fréquentes :
E : Enregistrement du fichier (s'il a été complété ou corrigé).
T : Tri aphabétique (par les mots anglais). C'est, avouons-le un petit luxe quasi inutile.
I : Impression du fichier sur imprimante, un autre gadget...
Q : Quitter le programme (= CLS : END).

— **Quand on est dans l'une des rubriques courantes, il suffit de taper Q et** ENTER pour revenir au menu. Ce retour est automatique pour les options E, T et I.

— En recherche "A" ou "F" **tout le fichier** est exploré de haut en bas à chaque demande, avec affichage des lignes-fiches satisfaisant la demande. Un bip sonore signale la fin de son travail. Pour 286 fiches, la durée est de 1,2 seconde...

— On peut demander le mot exact ou une **partie du mot** en faisant suivre d'un astérisque *.

Exemple :

Je demande BLOW UP ; il l'a, il affiche : n° 43 : BLOW UP = SE PLANTER. Si j'avais demandé BLOW-UP pas de réponse. Mais je l'aurais eu en demandant BLOW*. En demandant UP*, il m'aurait affiché les lignes de BLOW UP, SET UP, SUPPLY, etc... Vu ?

La recherche avec un tronçon de mot n'est pas plus lente qu'avec le mot intégral.

— La correction d'une fiche est très rapide. L'ancienne version apparaît à l'écran en bleu clair, vous écrivez en-dessous en jaune. Si l'un des deux mots n'est pas à corriger, faire ENTER quand le curseur est dessus, il sera conservé.

Il y a encore de nombreux petits détails sympathiques, mais étant de moindre intérêt, nous en parlerons plus loin.

Le fichier de l'auteur ne peut occuper les pages du "CPC", en revanche, il sera logeable sur la disquette bi-menseulle commercialisée par SORACOM. Vous pourrez alors le compléter à votre guise.

A présent, analysons un peu le listing.

LE DÉPART

— Lignes 50 à 70, c'est la parade habituelle contre le bug AMSTRAD de l'OPENOUT (voir CPC n° 3).

— Le tableau DIM M$(900,1) : (M$ = mots). Fixé très arbitrairement à 900, voire 999, il pourra par la suite être modifié bien au-delà. La colonne 0, c'est le mot anglais limité à une longueur de 14 ; la colonne 1, c'est la traduction française, chaîne limitée à 21 caractères.

— Ligne 250 : SUPER IMPORTANT ! Ce GOTO 1000 ne va servir que pour la **première fois** parce qu'il n'y a pas encore de fichier à lire. Dès que le fichier ANGLOFRA aura été enregistré (même avec quelques mots), il faudra effacer cette ligne ou intercaler un REM (apostrophe).
Si vous avez un lecteur de disquettes, le fichier sera sur la même face que le programme, en ce cas, vous ajouterez la ligne :

315 GOTO 400

— Ligne 350. CALL&BB06 stoppe le programme jusqu'à ce que l'on frappe une touche.

LE MENU

Toujours en ligne 1000 (mon "standard"), car en cas d'ennui il suffit de faire ESC et GOTO 1000.
Le nombre de mots NF apparaît en haut à droite de l'écran.
La ligne 1150 et le modèle 50000 sont de vieilles connaissances. Je rappelle que le module 50000 ("Réponse à un menu") est un sous-programme utilitaire enregistré que l'on joint au programme en cours d'écriture par MERGE "MENU".
En ligne 1160, GOSUB vers le sous-programme demandé, puis retour en 1000, ou fin du programme en 1520.

RECHERCHE DES MOTS

En ligne 2000 (option anglais) ou 2500 (option français) P est le numéro de la colonne du tableau DIM à explorer. Puis tout le travail se fait en GOSUB 3000.
En haut de l'écran, la fenêtre WINDOW# 1 sert à rappeler le mode d'emploi (l'astérisque et Q pour quitter) et à entrer le mot M$ à chercher. Celui-ci est mis en majuscules, puis on examine s'il est terminé par * ; en ce cas il en est dépouillé et traité en 3500 à l'aide de la fonction INSTR. Si pas d'astérisque, traitement par = . Dans les deux cas, on vérifie que la longueur autorisée n'est pas dépassée.
A chaque occurence rencontrée, la ligne réponse est affichée à l'écran. Scrolling normal lorsque l'écran est plein. Un bip sonore, CHR$(7) prévient de la fin de la recherche de 1 à NF

LA MODIFICATION (ou correction) - Ligne 5000

On met le flag MODIF = 1 puis renvoi en GOSUB 10000 : ce sous-programme est le plus gros puisqu'il est commun avec la saisie de nouveaux mots.
La fenêtre WINDOW#1 comporte quelques légendes et l'entrée du **numéro** de la fiche à corriger.
L'ancien contenu apparaît en bleu clair, le curseur se place d'abord sous le mot anglais que l'on peut retaper (en jaune), ENTER ; le curseur va ensuite sous le

mot français, idem. Généralement, l'un des deux mots n'est pas à modifier, taper alors ENTER pour le conserver.
Si on s'est trompé de numéro de fiche (cela arrive...), on presse donc deux fois la touche ENTER pour ne rien modifier. Il y a un CLS après chaque correction de fiche.

L'ENTRÉE DE NOUVEAUX MOTS - Ligne 10000

La fenêtre supérieure ne sert qu'à légender, ce qui laisse 19 lignes pour l'écran de saisie. Lorsque celui-ci est rempli, il n'y a pas de scrolling mais un CLS ; c'est dommage, mais je n'ai pas réussi à faire autrement, à cause de ces deux LINE INPUT sur la même ligne à la suite d'un PRINT USING.
La longueur de chaque mot entré est vérifiée ; s'il est trop long, il est tronqué. UPPER$ le met en majuscules.
SUPER ULTRA IMPORTANT ! Le LINE UP autorise l'entrée de divers signes de ponctuation tels que virgule, apostrophe, etc... mais n'entrez surtout pas de GUILLEMETS ! Il les accepte tant que le fichier est en RAM ; enregistrez-le et rechargez-le, et c'est le désastre à partir du premier guillemet (décalage). C'est un petit bug AMSTRAD à la lecture par la commande INPUT# 9.

LE TRI - Ligne 7000

C'est le banal "tri à bulle" par ordre alphabétique des mots anglais. Nous avons vu que le désordre ne peut ralentir la recherche des mots, alors à quoi sert-il ? A rendre le listing sur imprimante plus présentable ; en somme un gadget pour un autre gadget. En fait, il présente deux autres intérêts : le dépistage de doubles en examinant le listing. Une future et éventuelle extension du logiciel qui utilisera la "recherche par dichotomie". On en parlera en annexe...
Attention, un tri à bulle est très lent, cela peut faire des dizaines de minutes pour quelques centaines de fiches...

L'IMPRESSION - Ligne 8000

Vous avez déjà le listing de votre fichier de 300 mots, vous venez d'en ajouter une quinzaine : il est donc pratique que le programme vous demande : "à partir de quel numéro ?"
Pour soigner la présentation, le numéro est en PRINT USING et les mots sont tabulés. Une ligne en pointillés rejoint les mots anglais et français (fonctionnera sous SEIKOSHA-AMSTRAD).

PETITS CONSEILS DIVERS

— Pour un verbe anglais, ne mettez pas le TO.
— Entrez les mots au singulier.
— Si un mot anglais a deux sens, inscrivez-les dans le "mot " français en séparant par une virgule. Exemple :

PAD = BLOC, COMPLETER

— Où enregistrer le fichier ? Magnétophone : sur une cassette **séparée** et en **deux** endroits : compteur = 3 et compteur = 50. Lecteur de disquettes : sur la même face que le programme + sur une disquette de sécurité.

— En recherche, il est plus sécurisant d'utiliser l'astérisque.

LA RECHERCHE PAR DICHOTOMIE

Si plus tard le fichier atteint près de 2000 mots, chaque recherche durera près de neuf secondes ! Prohibitif ; d'où la méthode par dichotomie. Condition impérative : le fichier doit être **trié** . Soit le mot M$ à chercher. On le compare à celui du **milieu** du fichier (exemple : le n° 1000). Plus grand ou plus petit ? Plus grand. On considère donc la seconde moitié (1000 à 2000) et on compare M$ au mot milieu (n° 1500). Plus petit on compare alors M$ au mot n° 1250, etc. Dès que la tranche de localisation fait moins de 200 fiches, on procède à l'exploration fiche par fiche. Vous imaginez le gain en vitesse ?

```
10 'LEXIQUE ANGLAIS-FRANCAIS / AMSTRAD
20 'Michel Archambault / 1985
50 OPENOUT"BIDON"
60 MEMORY HIMEM-1
70 CLOSEOUT
200 MODE 1:BORDER 2:INK 2,20:PAPER 0:PEN 1:CLS
210 DIM M$(900,1):DEFINT A-Z
250 'GOTO 1000:'Ligne a effacer des que le fichier mots aura ete enregist
re.
300 'LECTURE FICHIER
310 CLS:PEN 3:LOCATE 9,4:PRINT "LEXIQUE ANGLAIS-FRANCAIS":PEN 1
320 LOCATE 3,8:PRINT "PREPAREZ LA CASSETTE OU DISQUETTE"
330 LOCATE 3,10:PRINT "CONTENANT LE FICHIER MOTS ANGLOFRA,"
340 PEN 2: LOCATE 10,15:PRINT "puis tapez une Touche...":PEN 1
350 CALL &BB06:CLS:PRINT:PRINT "  CHARGEMENT des MOTS ......":PRINT
400 OPENIN "ANGLOFRA":INPUT#9,NF
410 FOR I=1 TO NF:INPUT#9,M$(I,0),M$(I,1):NEXT
420 CLOSEIN
1000 'MENU PRINCIPAL
1010 CLS:INK 0,1:INK 1,24
1020 LOCATE 17,2:PEN 3:PRINT "M E N U":PEN 2:LOCATE 29,2:PRINT NF;"mots":
PEN 1
1030 LOCATE 6,4:PRINT "A = Anglais ---> Francais"
1040 LOCATE 6,6:PRINT "F = Francais --> Anglais"
1050 LOCATE 6,8:PRINT "C = Completer le lexique"
1060 LOCATE 6,10:PRINT "M = Modifications"
```

```
1070 LOCATE 6,12:PRINT "E = Enregistrement"
1080 LOCATE 6,14:PRINT "T = Tri alphabetique"
1100 LOCATE 6,16:PRINT "I = Impression"
1110 LOCATE 6,20:PRINT "Q = Quitter"
1150 TEX$="AFCMETIQ":GOSUB 50000
1160 ON K GOSUB 2000,2500,4000,5000,6000,7000,8000,1500
1170 GOTO 1000
1500 'FINAL
1510 CLS:LOCATE 10,12:PRINT "VOUS POUVEZ ETEINDRE.":PRINT:PRINT
1520 END
2000 'RECHERCHE ANGLAIS-->FRANCAIS
2010 R$="ANGLAIS":P=0
2020 GOSUB 3000
2030 RETURN
2500 'RECHERCHE FRANCAIS-->ANGLAIS
2510 R$="FRANCAIS":P=1
2520 GOSUB 3000
2530 RETURN
3000 'RECHERCHE DE TRADUCTION
3010 WINDOW #1,1,40,1,7:PAPER#1,0:PEN#1,1:CLS #1
3020 WINDOW #0,1,40,8,25:PAPER 2:PEN 0:CLS:PRINT
3030 LOCATE#1,4,2:PRINT#1,"Mot suivi de * = contenant ce mot"
3040 LOCATE #1,2,4:PRINT#1,"Q pour QUITTER:Le BIP=fin du fichier"
3050 PEN#1,3:LOCATE #1,3,6:PRINT#1,CHR$(18);"Mot ";R$;" ?: ";
3060 LINE INPUT #1,M$:M$=UPPER$(M$):IF M$="Q" THEN 3900
3080 IF RIGHT$(M$,1)="*" THEN 3500
3100 IF LEN(M$) >14+P*7 THEN 3050
3110 FOR I=1 TO NF
3120 IF M$=M$(I,P) THEN GOSUB 3800
3130 NEXT:PRINT CHR$(7);:GOTO 3050
3500 M$=LEFT$(M$,LEN(M$)-1)
3510 IF LEN(M$) >14+P*7 THEN 3050
3520 FOR I=1 TO NF
3530 IF INSTR(M$(I,P),M$) >0 THEN GOSUB 3800
3540 NEXT:PRINT CHR$(7);:GOTO 3050
3800 'AFFICHAGE LIGNE RESULTAT
3810 PRINT USING "###";I;:PRINT TAB(5);M$(I,0);TAB(18);":";M$(I,1)
3830 RETURN
3900 WINDOW#0,1,40,1,25:PAPER 0:PEN 1:CLS
3910 RETURN
4000 'COMPLETER LE LEXIQUE
4010 CLS
```

```
4020 LOCATE 14,4:PRINT "NOUVEAUX MOTS."
4030 LOCATE 16,10:PRINT "D'ACCORD ?":TEX$="ON":GOSUB 50000
4040 IF K=2 THEN 4060
4050 N=NF+1:GOSUB 10000:NF=N-1
4060 RETURN
5000 'MODIFICATION
5010 CLS:LOCATE 14,4:PRINT "MODIFICATION."
5020 LOCATE 16,10:PRINT "D'ACCORD ?":TEX$="ON":GOSUB 50000:IF K=2 THEN 59
00
5030 MODIF=1
5040 GOSUB 10000
5050 MODIF=0
5900 RETURN
6000 'ENREGISTREMENT
6010 CLS:LOCATE 2,5:PRINT "SAUVEGARDE SUR DISQUETTE OU CASSETTE"
6020 LOCATE 16,10:PRINT "D'ACCORD ?":TEX$="ON":GOSUB 50000
6030 IF K=2 THEN 1000
6080 OPENOUT "ANGLOFRA"
6090 WRITE #9,NF:FOR I=1 TO NF:WRITE #9,M$(I,0),M$(I,1):NEXT
6100 CLOSEOUT
6900 RETURN
7000 'TRI
7010  CLS:LOCATE 13,4:PRINT "TRI DU FICHIER:":LOCATE 16,11:PRINT "D'ACCOR
D ?":TEX$="ON":GOSUB 50000:IF K=2 THEN 7900
7020  LOCATE 16,21:PRINT "Patience...":T=0
7060  F=0:FOR J=1 TO NF
7070 IF M$(J,0) >= M$(J-1,0) THEN 7090
7080  FOR R=0 TO 1:M$(0,R)=M$(J,R):M$(J,R)=M$(J-1,R):M$(J-1,R)=M$(0,R):NE
XT:F=1
7090 NEXT
7100 IF F THEN 7060
7900 RETURN
8000 'IMPRESSION
8010 CLS:LOCATE 5,10:PRINT "L'IMPRIMANTE EST-ELLE PRETE ?"
8020 TEX$="ON":GOSUB 50000:IF K=2 THEN 8900
8030 CLS:LOCATE 6,10:INPUT "A PARTIR DE QUEL NUMERO ? ",D$
8040 D=VAL(D$):IF D=0 OR D>NF THEN 8030
8050 PRINT #8,CHR$(27);CHR$(64):FOR I=D TO NF
8060 PRINT #8,USING "###";I;:PRINT #8," ";M$(I,0);STRING$(16-LEN(M$(I,0))
,".");" ";M$(I,1)
8070 NEXT:PRINT #8
8900 RETURN
```

```
10000 ' SAISIE
10020 WINDOW#0,1,40,7,25:PAPER 0:PEN 1:CLS
10030 WINDOW#1,1,40,1,5:PAPER#1,2:PEN#1,3:CLS#1
10040 LOCATE #1,2,2:PRINT #1,"Pour quitter ( Menu ) --> Q + ENTER"
10050 LOCATE #1,2,4:PRINT #1,"N";SPC(5):"ANGLAIS";SPC(11);"FRANCAIS"
10060 L=0
10070 IF MODIF THEN GOSUB 10500:GOTO 10200
10100 L=L+1:LOCATE 1,L:PRINT USING "###":N
10105 LOCATE 5,L:LINE INPUT "",M$(N,0)
10110 M$(N,0)=UPPER$(M$(N,0)):IF M$(N,0)="Q" THEN 10200
10120 IF LEN(M$(N,0)) >14 THEN M$(N,0)=LEFT$(M$(N,0),14)
10130 LOCATE 18,L:LINE INPUT ":",M$(N,1)
10140 M$(N,1)=UPPER$(M$(N,1)):IF LEN(M$(N,1)) >21 THEN M$(N,1)=LEFT$(M$(N
,1),21)
10150 N=N+1:IF L=19 THEN CLS:L=0
10160 GOTO 10070
10200 WINDOW#0,1,40,1,25:CLS
10300 RETURN
10500 'FICHE A MODIFIER
10510 PEN 2:LOCATE 3,1:INPUT "QUEL NUMERO ?( Q pour Quitter ):",N$:N$=UPP
ER$(N$)
10520 IF N$="Q" THEN PEN 1:GOTO 10700
10530 N=VAL(N$):IF N=0 OR N>NF THEN 10510
10540 LOCATE 1,3:PRINT USING "###";N;:PRINT TAB(5);M$(N,0);TAB(19);M$(N,1
)
10550 PEN 1
10560 LOCATE 5,5:LINE INPUT "",M$:M$=UPPER$(M$)
10570 IF M$="" THEN 10600
10580 IF LEN(M$) >14 THEN M$=LEFT$(M$,14)
10590 M$(N,0)=M$
10600 LOCATE 19,5:LINE INPUT "",M$:M$=UPPER$(M$)
10610 IF M$="" THEN 10690
10620 IF LEN(M$) >21 THEN M$=LEFT$(M$,21)
10630 M$(N,1)=M$
10690 CLS:GOTO 10510
10700 RETURN
50000 'REPONSE A UN MENU
50010 LT=LEN(TEX$):R$=""
50020 LOCATE 15-LT,24:PRINT "Reponse (";
50030 FOR I=1 TO LT-1
50040 PRINT MID$(TEX$,I,1);",";:NEXT
```

```
50050 PRINT RIGHT$(TEX$,1);")";CHR$(154);CHR$(243);CHR$(207)
50060 TEX$=UPPER$(TEX$)
50070 WHILE R$="":R$=INKEY$:WEND
50080 R$=UPPER$(R$):K=INSTR(TEX$,R$)
50090 IF K=0 THEN R$="":PRINT CHR$(7);:GOTO 50070
50100 RETURN
64535 '------- FIN DE LISTING ---------
```

4

TURBO PASCAL

Denis BOURQUIN

L'AMSTRAD possède déjà un compilateur PASCAL, mais ce compilateur fonctionne sous le système d'exploitation Basic et est de taille réduite ; il permet simplement une bonne initiation au langage PASCAL, mais il lui manque surtout les opérations sur les fichiers. Il manquait donc à l'AMSTRAD un compilateur performant et fonctionnant sous CP/M, il existe maintenant et nous ne pouvons espérer mieux.

En effet, un des meilleurs compilateurs PASCAL : TURBO PASCAL est maintenant disponible sur AMSTRAD. Tous les programmeurs s'intéressant au langage PASCAL ont entendu parler de TURBO PASCAL.
Pour ceux qui ne connaîtraient pas le langage Pascal, nous allons en dire quelques mots.
Le langage Pascal a été développé en 1971 par le Pr. Niklaus WIRTH à l'université technique de Zurich ; ce langage fut surtout utilisé, au début de son existence, dans les universités, mais depuis le début des années 80, il connaît un grand essor et est utilisé dans de nombreuses applications professionnelles. C'est un langage mettant en application la méthode de programmation appelée programmation structurée.

Un Français qui étonne l'Amérique

Mais avant de parler de ce qui a fait le succès de TURBO PASCAL, nous dirons quelques mots de son concepteur qui est français et s'appelle Philippe KAHN. Il a fondé, aux États-Unis, une société : BORLAND INTERNATIONAL. La qualité et les prix ridiculement bas de ses logiciels ont fait la force et la renommée de cette société. Pour s'en rendre compte, il suffit de regarder le prix du compilateur TURBO PASCAL sur AMSTRAD : 741,25 F, alors que jusqu'à ces dernières années le prix d'un compilateur Pascal souvent moins performant était de 10 à 20 fois supérieur.

Un compilateur Turbo

Tout le monde connaît l'intérêt d'un langage compilé, mais regrette souvent le temps passé sous des éditeurs peu performants puis en compilation pour s'apercevoir qu'il restait une erreur et qu'il fallait revenir sous l'éditeur, rechercher cette erreur et recompiler. Comme nous le verrons par la suite, TURBO PASCAL rend aisées toutes ces tâches ingrates. Mais faisons un essai de ce fameux compilateur turbo.

TURBO PASCAL est un compilateur qui tourne sous CP/M ; nous passons donc sous CP/M et regardons ce que contient la disquette TURBO PASCAL. Nous y trouvons un fichier TURBO.COM qui est le compilateur-éditeur, un programme d'installation du terminal propre à l'AMSTRAD, un fichier TURBO.MASG qui contient les messages d'erreurs, un fichier TURBO.OVR propre à la version CP/M et qui permet l'exécution de programme. COM sous TURBO comme des utilitaires de copie ou tout autre programme, un certain nombre de fichiers.PAS, qui sont des fichiers source de programmes PASCAL de démonstration que nous avons beaucoup appréciés, et enfin un fichier READ.ME qui est assez explicite par son nom. Ce dernier est un fichier texte qui contient un certain nombre de conseils et d'explications quant à l'installation et l'utilisation de TURBO PASCAL sur AMSTRAD.

TURBO PASCAL est un compilateur qui contient son propre éditeur de texte ; nous allons donc appeler TURBO pour écrire notre premier programme PASCAL. Les premiers messages s'affichent, nous voyons que nous sommes en possession de la version 3.00A de TURBO PASCAL pour CP/M-80,280. Et nous avons le message : INCLUDE ERROR MESSAGES (Y/N)?. Si nous répondons Y, nous avons un menu qui s'affiche, ainsi que la quantité d'octets mémoires disponibles, mais voyons la figure 1, exemple d'écran que nous allons expliquer.

Sur cet écran apparaît la liste des commandes disponibles, et l'espace mémoire disponible pour l'écriture de nos programmes : 6637 octets ; la première impression est que cet espace est plus que réduit sous CP/M. Il est possible d'augmenter sa taille en n'incluant pas les messages d'erreur ; dans ce cas, l'espace disponible est de 7952 octets. En effet, sous CP/M, il reste environ 39 k à la disposition de l'utilisateur, le compilateur TURBO en occupant 31 k, il ne nous reste donc pour loger nos sources et nos objets que 8 k. Mais rassurez-vous, avec TURBO PASCAL, il est possible de faire des programmes beaucoup plus grands que 8 k-octets. Notons que, dans ce cas, nous perdrons les avantages de rapidité de TURBO PASCAL. Les programmes correspondant à des applications importantes seront compilés dans un fichier .COM, pour être exécutés et, dans ce cas, ils pourront aller jusqu'à 39 k-octets de code. Il faudra prendre la précaution de diviser le source en plusieurs fichiers qui seront inclus à un fichier principal lors de la compilation, car il ne peut y avoir plus de 8 k-octets de source présents en mémoire. Et si votre application ne tient pas sur 39 k-octets, il vous reste encore la possibilité de scinder en overlay, car TURBO supporte aussi cette technique de programma-

tion. Nous venons de voir un exemple d'écran qui résume les commandes disponibles sur TURBO. Ces commandes sont accessibles par une lettre en majuscule.

L : permet de changer l'unité de disque de travail.
W : permet de changer le fichier de travail.
M : permet de spécifier un fichier principal, ce dernier peut-être différent du fichier de travail.
E : permet de passer sous l'éditeur incorporé à TURBO PASCAL
C : lance la compilation dans le mode choisi par Option.
R : permet d'exécuter un programme compilé ou si le programme n'a pas été compilé, le compile avant de l'exécuter.
S : permet de sauvegarder le source en cours.
X : permet d'exécuter un autre programme en cours d'utilisation de TURBO
D : permet de faire un directory de disque sous TURBO.
Q : retourne au système d'exploitation CP/M
O : permet de choisir entre trois options, M, C et M. M est le code mémoire par défaut, le code objet se trouve en mémoire près la compilation C : le code objet issu de la compilation est écrit sur un fichier disque.COM, H : le code objet est écrit dans un fichier chaîné ; ce fichier ne contient pas la librairie Pascal et sera donc appelé par un autre fichier.

```
Logged drive: A

Work file:

Main file:

Edit        Compile    Run        Save

eXecute     Dir        Quit       compiler Options



Text: 0 bytes (8113-8118)

Free: 6637 bytes (8119-9B06)
```

Mais passons tout de suite à un essai de l'éditeur. Nous choisissons l'option E dans le menu, une question apparaît : "Work file ?" Entrons essai.pas, l'écran s'efface, et nous sommes sous l'éditeur.

Nous sommes en présence d'un éditeur pleine page très performant ; les déplacements à l'intérieur d'un texte sont obtenus à partir du pavé des flèches de L'AMSTRAD, et il est possible de se déplacer d'un caractère, d'un mot,

d'une ligne ou d'une page. Cet éditeur peut effectuer l'indentation automatique, il peut recadrer chaque mot d'une ligne par rapport à la précédente, ce qui favorise la lisibilité du programme. A ces commandes s'ajoutent toutes les commandes d'insertion, suppression, déplacement, remplacement. Mais il n'est pas nécessaire de faire l'éloge de tels éditeurs qui sont bien plus agréables d'utilisation que les éditeurs ligne, et en plus quand ils sont bien faits, ça ne peut qu'augmenter leur intérêt.

Regardons maintenant la syntaxe de ce PASCAL. Il apparaît au premier coup d'œil jeté dans le manuel fourni avec la disquette qu'il suit de la définition du PASCAL STANDARD. Il est à noter que ce livre est rédigé en français et est très bien conçu ; il contient tous les renseignements nécessaires à l'utilisation de TURBO PASCAL. Le débutant en PASCAL a ainsi à sa disposition une excellente introduction à PASCAL avec le livre et la disquette TURBO TUTOR, qui sont vendus séparément.

TURBO PASCAL comprend un certain nombre d'extensions par rapport au PASCAL STANDARD, ce sont :

— les variables du type adresse absolue,
— la manipulation de bits et d'octets,
— l'accès direct à la mémoire et aux ports d'entrées-sorties (la mémoire est considérée comme un tableau prédéclaré MEM de 65536 octets, il en va de même pour les entrées-sorties qui sont le tableau PORT de 256 octets).
— les chaînes de caractères dynamiques,
— l'ordre des sections dans la partie déclaration des programmes est libre,
— la possibilité d'accéder aux fonctions du système d'exploitation par des procédures et fonctions prédéfinies : BDOS, BIOS, BDOSHL, BIOSHL,
— la possibilité de 4 fichiers imbriqués,
— la possibilité de générer du code machine en ligne par l'instruction INLINE,
— les opérations logiques peuvent se faire sur des entiers,
— la possibilité de chaîner des programmes avec variables communes,
— les fichiers de données à accès aléatoire,
— les constantes stucturées,
— les fonctions de conversion de type.

Nous avons relevé comme autre particularité les instructions BLOCKREAD et BLOCKWRITE qui servent aux manipulations de fichier sans type. Ces procédures travaillent sur des enregistrements de 128 octets.

L'accès au multi-fenêtrage de l'AMSTRAD peut se faire par l'envoi de la valeur 26 à l'écran et des coordonnées de la fenêtre. Exemple : si ces coordonnées sont X1, X2, Y1, Y2 et ont été déclarées BYTE, le positionnement dans cette fenêtre se fera par : WRITE (#26,CHR (X1—1), CHR(Y1—1), CHR(X2—1), CHR (Y2—1)).

L'utilisation des variables dynamiques se fait au travers des procédures standards NEW et DISPOSE ainsi que les procédures MARK et RELEASE.

TURBO PASCAL permet les manipulations de chaîne et possède de nombreuses procédures pour cela : CONCAT, COPY, DELETE, INSERT, LENGTH, POS, STR, VAL.

En résumé, avec TURBO PASCAL, nous avons un compilateur PASCAL très complet. Notons que pour l'utilisation des possibilités graphiques de

l'AMSTRAD, TURBO PASCAL, dans sa version CP/M, ne contient aucune fonction ou procédure spécifique.
Nous avons essayé d'utiliser les routines en ROM, mais ce fut sans grand succès. Rassurez-vous, il est prévu avec TURBO pour AMSTRAD un pack graphique qui permettra d'utiliser toutes les ressources de l'ordinateur, mais lorsque nous avons écrit ces lignes, nous ne le possédions pas ; lorsque nous l'aurons reçu, nous en dirons quelques mots.

Après avoir passé en revue la syntaxe, voyons ce qui fait aussi la force de ce Pascal : sa rapidité de compilation. Il n'est pas nécessaire de dire que le temps de compilation en mémoire est négligeable. Lorsqu'une erreur est rencontrée au cours de la compilation, il suffit d'appuyer sur la touche ESC pour retourner instantanément sous l'éditeur avec le curseur situé juste derrière cette erreur ; il n'y a plus qu'à la corriger et relancer la compilation. La mise au point d'un programme est aussi aisée qu'en Basic. La compilation de programmes importants se fait sur disque dans un fichier.COM. Sur la disquette TURBO se trouve un programme de démonstration qui n'est rien d'autre qu'un tableur. Ce programme occupe 1200 lignes et est compilé en 30 secondes. Il permet de mettre en évidence les possibilités infinies de TURBO PASCAL.
Nous avons apprécié les programmes de démonstration livrés avec la disquette tels que le minitableur, la commande lister, l'utilisation des fenêtres, l'utilisation de la ligne de commande.
L'AMSTRAD, grâce à son système d'exploitation CP/M, a pu se doter là d'un des meilleurs compilateurs Pascal actuels.

4

JOUER AVEC LES INTERRUPTIONS

AMSTRAD est capable de faire plusieurs choses à la fois, et ce travail en "multitâche" est accessible, vous le savez, depuis le Basic. EVERY et AFTER n'étant pas universels, nous avons cherché à aller plus loin :

Afficher l'heure en Basic sous le contrôle d'EVERY est extrêmement simple, seulement il est impossible d'inclure un tel sous-programme dans un programme utilisant des INPUT : la pendule se bloque pendant la durée de l'entrée des variables (elle est remise à l'heure ensuite). Devant un tel problème, on est conduit à utiliser le langage machine.

L'AMSTRAD est capable de traiter les interruptions, synchrones ou asynchrones, auxquelles on peut associer des "événements"exécutables cycliquement. Plusieurs éléments sont paramétrables et, sans entrer en profondeur dans les détails, nous vous proposons, au moyen de deux petits exemples d'application, de faire plus ample connaissance avec ces interruptions.

Deux types d'interruptions nous intéressent :

— FAST TICKER apparaissant tous les 1/300e de seconde,

— FRAME FLYBACK, tous les 1/50e de seconde, correspondant au retour du spot assurant le balayage.

Les équivalents "soft" de ces interruptions sont appelés "EVENTS" et, si leur manipulation s'avère un peu délicate au début, vous conviendrez que savoir les utiliser ouvre pas mal d'horizons. L'installation d'un "EVENT" suit un schéma relativement simple. L'événement est décrit par sa classe, sa priorité, et l'adresse de la routine qui est appelée pour le traiter.

La classe : événements synchrones ou asynchrones. Les événements synchrones sont traités immédiatement, les synchrones dès que le système a le temps. Pour les événements synchrones, on attribue une classe de priorité. Quant à l'adresse de la routine, elle est donnée au système lors de la "description" initiale de l'événement.

A chaque "EVENT" est attribuée une zone mémoire (d'un certain nombre d'octets en fonction du type d'interruption) réservée au système.

L'initialisation d'un EVENT se fait comme suit :
HL contient l'adresse de la zone mémoire réservée.
B contient "la classe".

Bit	7	6	5	4 3 2 1	0
1	Asyn	Express Ø		niveau priorité	adresse proche
Ø	Sync	Normal Ø			adresse lointaine

C contient une valeur fonction de l'adresse mémoire de la routine d'événement (selon qu'elle est en RAM, ROM, ROM externe..).
DE contient l'adresse de la routine qui sera exécutée à chaque interruption.
Voici un premier exemple d'application : remplacer le curseur fixe par un curseur clignotant. On va donc, successivement, éteindre et allumer le curseur.

On utilisera pour cela le FRAME FLYBACK (donc toutes les 20 ms). La zone mémoire à réserver pour le système est de 9 octets. Le type d'événement est "asynchrone". La routine sera en RAM centrale. Le contexte a été sauvegardé à chaque exécution de la routine interruption : bien que cela ne soit pas nécessaire, notre auteur Eddy DUTERTRE a pris toutes ses précautions. Les routines en BB7B et BB7E correspondent respectivement à l'autorisation et l'interdiction d'affichage du curseur.
Si le sujet (interruptions) traité ici rapidement, vous passionne, faites-le nous savoir, et nous ferons un article plus complet sur la question.

```
10000 MEMORY &7FFF
10010 FOR n=&8000 TO &8029:READ d$:POKE n,VAL("&"+d$):NEXT
10020 '
10030 CALL &8000:NEW
10050 DATA 21,2A,80,06,81,0E,00,11,0E,80
10060 DATA CD,D7,BC,C9,C5,D5,E5,F5,3A,FF
10070 DATA 7F,E6,10,28,05,CD,7B,BB,18,03
10080 DATA CD,7E,BB,21,FF,7F,34,F1,E1,D1
10090 DATA C1,C9
```

PAGE 1

```
8000 212A80                LD    HL,L802A    Adresse du bloc EVENT.
8003 0681                  LD    B,81H       Type asynchrone en mémoire centrale.
8005 0E00                  LD    C,0         Mémoire centrale, donc pas en ROM externe.
8007 110E80                LD    DE,L800E    Adresse de la routine à exécuter.
800A CDD7BC                CALL  0BCD7H      Initialise le bloc d'EVENT.
800D C9                    RET
800E C5          L800E:    PUSH  BC        } Sauvegarde du contexte avant traitement de
800F D5                    PUSH  DE        } l'interruption.
8010 E5                    PUSH  HL        }
8011 F5                    PUSH  AF        }
8012 3AFF7F                LD    A,(7FFFH)
```

```
8015 E610                AND  10H
8017 2805                JR   Z,L801E     Effectue le clignotement (cadence) en fonc-
8019 CD7BBB              CALL 0BB7BH      tion du contenu de 7FFF.
801C 1803                JR   L8021
801E CD7EBB    L801E:    CALL 0BB7EH
8021 21FF7F    L8021:    LD   HL,7FFFH
8024 34                  INC  (HL)
8025 F1                  POP  AF    )
8026 E1                  POP  HL    |
8027 D1                  POP  DE    }     Restitue le context initial.
8028 C1                  POP  BC    |
8029 C9                  RET        )
802A 00        L802A:    NOP
```

4

DES COULEURS PLUS CLAIRES

Michel ARCHAMBAULT

INK... PEN... PAPER... BORDER...Bouh, que c'est compliqué ! Expliqué tout autrement, cela va devenir enfin clair :

Il y a cinq notions différentes :

— **Le numéro de référence de la teinte** ; Il y en a 27 numérotées de 0 (noir) à 26 (blanc brillant). C'est la liste de la page E3.2 du manuel d'origine. Appelons-le RT, c'est-à-dire référence teinte.

— **La palette de teintes que l'on se sélectionne, c'est INK.** En mode normal, mode 1, on n'a droit qu'à quatre INK. On les numérote de 0 à 3. Dès la mise sous tension (MODE 1), on nous propose les INK suivants :
INK 0,1 : bleu parce que RT = 1 = bleu,
INK 1,24 : RT = 24 c'est jaune vif,
INK 2,20 : RT + 20, un turquoise vif,
INK 3,6 : RT = 6 = rouge vif.
Vous voulez remplacer le turquoise par le vert de RT = 9 ? Facile, faites INK 2,9 (en mode direct ou dans un programme). En MODE 0, on a 16 INK sur la palette, de INK 0 à INK.15. Les quatre premiers (0 à 3) sont les mêmes RT que précédemment. A quoi correspondent-ils ? Consultez le tableau de la page E3.4 du manuel. Vous voulez du noir (noir RT = 0) ? C'est-INK n° 5
En MODE 2, on n'a que 2 INK, INK 0 et INK 1.

— **PAPER**, c'est la couleur de fond. Par défaut, l'amstrad prend pour PAPER l'INK 0, bleu foncé. Vous voulez un fond rouge ? Faites PAPER 3, parce que INK 3, c'est le rouge sur notre palette. Si vous faites PAPER 0,3, il y aura clignotement bleu-rouge (aucune rétine n'y résiste...).

— **PEN** (stylo en anglais). C'est la couleur des caractères, des traits. Même syntaxe que PAPER, on le fait suivre du numéro d'INK. Par défaut, c'est INK 1 (jaune vif). Vous voulez des lettres bleues sur fond rouge ? Faites PAPER 3 : PEN 0. On peut écrire PEN avant PAPER, peu importe.

— **BORDER**, c'est la couleur de la bordure, elle est totalement **indépendante** des couleurs présentes sur la palette, car on fait suivre BORDER, non pas d'un

numéro INK, mais du numéro RT, désiré (de 0 à 26). En MODE 1, on peut avoir 4 teintes en INK plus une cinquième en BORDER.

Exemple, BORDER 16 = orange clair.

OK ? Bon, alors voyons quelques cas particuliers.

— Par étourderie PAPER et PEN ont le même numéro d'INK ; pire qu'un CLS parce qu'il n'y a pas de "READY". Ne vous affolez pas, pressez ESC deux fois (Break) puis tapez PEN 1 (ou 2 ou 3) et ENTER, et votre texte réapparaît.

— En MODE 1, faisons PEN 6, alors que l'on ne devrait pas dépasser PEN 3 ; pas de message d'erreur, l'AMSTRAD repart à INK 0 : PEN 3 = PEN 0 : PEN 5 = PEN 1 : PEN 6 = PEN 2 ; etc... jusqu'à PEN 26.

— Pour modifier les couleurs. Rappelez-vous les deux choses prises par défaut lors de la mise sous tension, à savoir la palette des INK/n° RT (page E3.4) et le fait que PAPER = INK 0 et PEN = INK 1, ou de dire PEN 2 ou PAPER 3, ou de combiner ces deux techniques.

Attention ! Il y a des mariages malheureux entre deux couleurs voisines ; perte de netteté (l'AMSTRAD n'y est pour rien, le moniteur non plus), ou contraste trop faible ou insoutenable. A vous de chercher quelques bons accords.

KILOBYTE

Philippe PEREZ
D. CARDON

Le chapitre cher aux paresseux mais aussi à ceux qui débutent et ont envie de découvrir le clavier sans taper des kilo... mètres de listing.

Septembre, la fin des vacances ! Conservez le souvenir de la mer et du cri des mouettes avec le programme de D. CARDON. A écouter en fermant les yeux.

Philippe PEREZ, lui, vous propose un petit utilitaire permettant de redéfinir facilement les touches du pavé numérique.
Vous pouvez leur affecter les mots-clés les plus utilisés, en vue de saisir un long listing.
La phase "affectation" étant achevée, vous pourrez vérifier qu'il n'y a pas d'erreur.
L'appui sur "F" permet de quitter le programme.

```
10 '----- Bruit de vagues ----- 19/5/85
---------- D.Cardon -----------------
20 ENV 1,1,5,1,10,1,10,15,-1,50:c=1
30 WHILE INKEY$="":tps%=RND*400:va%=RND*
21:mou%=RND*2
40 SOUND c,0,300+tps%,0,1,,10+va%:c=3-c:
IF mou%=1 THEN GOSUB 100
50 WEND:CALL &BCB6:END
100 '----- Mouette ----- 20/5/85 ----- D
.Cardon ------------------------------
110 ENV 2,1,RND*(7+1),20
120 FOR j=1 TO 2:at%=RND*100:h2%=RND*5
130 FOR i=1 TO 23
```

```
140 READ d%,h%,v%
150 SOUND 4,h%+h2%,d%,v%,2:NEXT:SOUND 4,
0,15+at%:RESTORE
160 NEXT:RETURN
170 DATA 1,70,1,2,69,4,1,68,5,1,67,5,1,6
6,6,1,65,6,1,60,7,2,55,7,3,52,7,4,51,7,5
,50,7,18,48,6,2,50,7,1,51,6,1,52,5,1,53,
5,1,54,4,1,55,4,1,57,3,1,58,3,1,59,2,1,6
0,1,3,65,1
```

```
10 MODE 0:INK 1,1:INK 0,15
20 LOCATE 1,10:PRINT "    REDEFINITION

              DU

                                    PAVE NUMERIQUE"

30 FOR I=1 TO 400 STEP 10:PLOT 1,I:DRAW
640,I,1:NEXT
40 BORDER  4,12
50 FOR I=1 TO 600:SOUND 1,3932/I,1,5:NEX
T
60 BORDER 1
70 MODE 2:N=-3
80 FOR J=1 TO 4:READ A,C
90 FOR I=48 TO 480  STEP 192
100 N=N+1:PLOT I,A
110 DRAW I+48,A:DRAW I+48,C
120 DRAW I,C:DRAW I,A
130 PLOT I-4,A+4:DRAW I+52,A+4
140 DRAW I+52,C-4:DRAW I-4,C-4:DRAW I-4,
A+4
150 LOCATE (I+24)/8,(C+32)/16:PRINT N
160 NEXT I,J:DATA 368,320,272,224,176,12
8,80,32
170 LOCATE 9,22:PRINT " 0"
180 LOCATE 33,22:PRINT " ";CHR$(144)
190 LOCATE 55,22:PRINT "ENTER.":L=128
200 FOR I=1 TO 12:READ F,G
210 LOCATE F,G:INPUT V$
220 KEY L,V$+CHR$(32):L=L+1:NEXT
230 DATA 15,22,15,16,39,16,63,16,15,10,3
```

```
9,10,63,10,15,4,39,4,63,4,39,22,63,22
240 LOCATE 29,25:PRINT "(* * F = F I N *
 *)"
250 WINDOW #1,5,75,13,13:PAPER #1,1:PEN
#1,0:CLS #1
260 INPUT #1,"     ESSAIS : ";E$
270 IF E$="F" OR E$="f" THEN 290
280 CLS #1:GOTO 260
290 MODE 1
```

4

TRUCS ET ASTUCES

J.M. LAQUITTANT
50250 DONVILLE DES BAINS

Dans votre rubrique Banc d'Essai des Logiciels du n° 3, vous présentez MULTIGESTION. Contrairement à ce que vous annoncez, il est possible de titrer les rubriques. Il faut valider le curseur clignotant avec COPY et entrer le titre désiré dans les 18 postes utilisables. Le report est de plus automatique de l'option "Comptabilité journalière" vers l'option "Budget".

M. ROUSSELET
95220 HERBLAY

Pour une fois que le Basic s'avère plus intéressant que l'assembleur, je ne résiste pas au plaisir de donner une suite au programme CIRCLE d'Eddy DUTERTRE, paru dans CPC n° 3. Mon petit programme ci-dessous permet de tracer un cercle quelconque en 1 à 2 secondes, ce qui est assez rapide.
Le principe : au lieu d'utiliser les équations paramétriques classiques X = Yc + R cos t et Y = Yc + R sin t,on détermine une suite de points du cercle où les coordonnées de chaque nouveau point se déduisent de celles du précédent. Les paramètres 0.01, 0.141, 0.99 et 45 ont été choisis empiriquement.

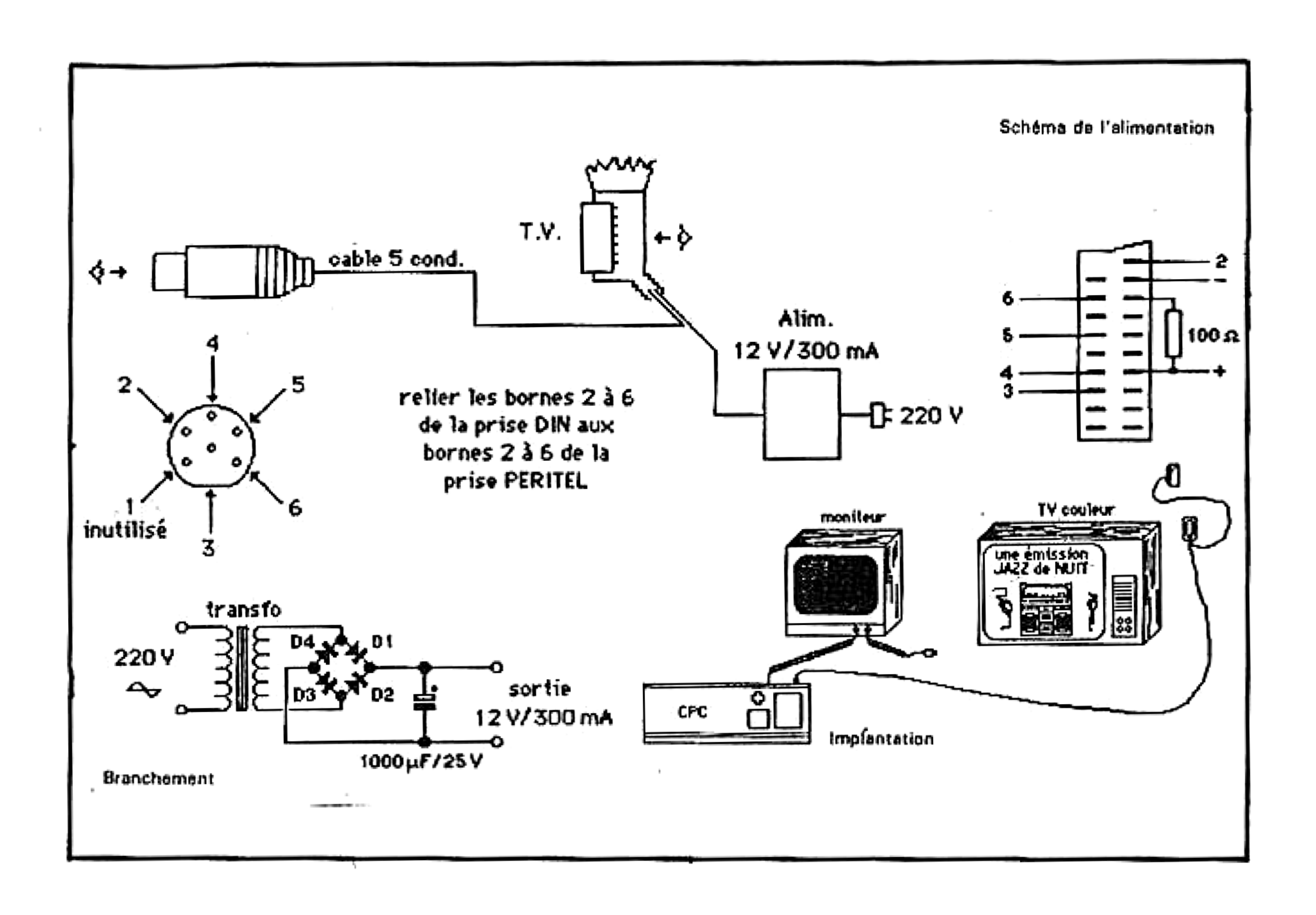
```
10 INPUT "ABSCISSE DU CENTRE   :";XC
20 INPUT "ORDONNEE DU CENTRE   :";YC
30 INPUT "RAYON DU CERCLE      :";R
40 X1=XC+R : Y1=YC
50 E=0.01*XC+0.141*YC : F=0.01*YC-0.141*
XC
60 FOR N=1 TO 45
70 X2=E+0.99*X1-0.141*Y1 : Y2=F+0.99*Y1+
0.141*X1
80 PLOT X1,Y1 : DRAW X2,Y2
90 X1=X2 : Y1=Y2
100 NEXT N
```

Imprimante peut économe

Si vous utilisez une imprimante autre que la DMP 1 sur votre AMSTRAD, vous avez peut-être constaté qu'elle avait une fâcheuse tendance à laisser une ligne blanche entre deux lignes imprimées. Certains préconisent de modifier par soft (cela est possible sur certaines imprimantes) l'espacement entre les lignes ; c'est bien, mais c'est aussi une manière de masquer le défaut. La solution passe par l'examen de la description du connecteur CENTRONICS : on s'aperçoit que, si la broche 14 est forcée à la masse, l'imprimante effectuera un saut de ligne automatique après chaque impression (AUTO LINE FEED). C'est l'origine du défaut car l'AMSTRAD envoie une masse sur ce point (toutes les liaisons "non utilisées" par l'interface AMSTRAD étant, à tort, fixées à la masse). Le remède consiste donc à isoler ce point 14, au moyen d'un minuscule adhésif, ou à couper le fil correspondant sur la nappe, mais là, gare ! ne vous trompez pas...

Christian HUGEL
68170 RIXHEIM

Voici comment réaliser un miracle avec très peu de composants : avoir la couleur en possédant un AMSTRAD monochrome. Naturellement, vous aurez compris qu'il ne s'agit pas de transformer le moniteur, mais plus simplement de réaliser un cordon qui raccordera la console au téléviseur couleur familial, pourvu que celui-ci soit doté d'une prise Péritel. Le coût de la réalisation est extrêmement bas, même en tenant compte de l'alimentation 12 volts dont nous donnons un shéma. Cette alimentation permet de déconnecter la partie réception HF du téléviseur. Elle est indispensable car, l'AMSTRAD fonctionnant sous 5 volts, ne permet pas de commander la commutation via le câble.
Nomenclature des composants
— 1 connecteur Péritel
— 1 prise DIN mâle 6 broches à 60°
— 3 mètres de câble à 5 conducteurs
— 1 résistance 100 ohms (brun-noir-brun-or)
— 1 alimentation 12 V/500 mA type calculatrice
ou
— 1 transformateur 220 V/12 V
— 4 diodes 1N4001 ou 1N4007
— 1 condensateur 1000 uF/35V


Schéma de l'alimentation
cable 5 cond.
T.V.
Alim.
12 V/300 mA
220 V
100 Ω
inutilisé
relier les bornes 2 à 6
de la prise DIN aux
bornes 2 à 6 de la
prise PERITEL
moniteur
TV couleur
une émission
JAZZ de NUIT
transfo
220 V
D4
D1
D3
D2
sortie
12 V/300 mA
1000 µF/25 V
CPC
Implantation
Branchement

KILOBYTE

Imaginez que votre AMSTRAD vient de tomber en panne : il ne lui reste plus qu'un petit kilo-octet de mémoire. Saurez-vous écrire des programmes originaux performants qui ne tiennent que dans ce kilo-octet ? Attention ! vos programmes devront être écrits en Basic, mais les routines en code machine présentées sous forme de data sont acceptées. Faites-nous parvenir vos progammes sur cassette, accompagnés d'une brève explication. Les cassettes vous seront retournées si vous le demandez.
Alors, tous à vos claviers. Il faut réussir à loger le maximum d'intelligence dans un kilo-octet. Les meilleurs programmes seront mis à l'honneur dans CPC et leurs auteurs récompensés.
Deux fainéants astucieux pour KILOBYTE ce mois-ci.
Erik BAJART nous propose d'apprendre le MORSE en traduisant les textes entrés en codes visuel et auditif. Si ce n'est pas la meilleure méthode pour débuter, elle pourra peut-être susciter des vocations.
Michel DUBOIS, quant à lui, préfère faire réviser aux enfants leurs tables de multiplication, tout en faisant travailler la mémoire. Savant mélange de 7 x 5 = et du jeu du "retourné", le tout en quelques lignes. Au travail !

```
10 BORDER 5:MODE 1:PRINT:DEFINT a-z:s=12
5:ENV 1,15,1,1:IF HIMEM>43900 THEN SYMBO
L AFTER 95:SYMBOL 95,0,0,0,0,0,&7E,&7E
20 PRINT,"*** MORSE ***":a$="0123456789A
BCDEFGHIJKLMNOPQRSTUVWXYZ.,:?'-()":C$="
_OGCA@FX\^%8:,"+CHR$(34)+"2.0$7-4'&/6=*(
#)1+9;<5SX,>AMM":PRINT:PRINT:PRINT:INPUT
"DUREE D'UNE NOTE (5 a 30) ";d:PRINT:PRI
NT"Ecrivez le message :":PRINT:LINE INPU
T m$
30 m$=UPPER$(m$):PRINT:PRINT:e=53:GOSUB
```

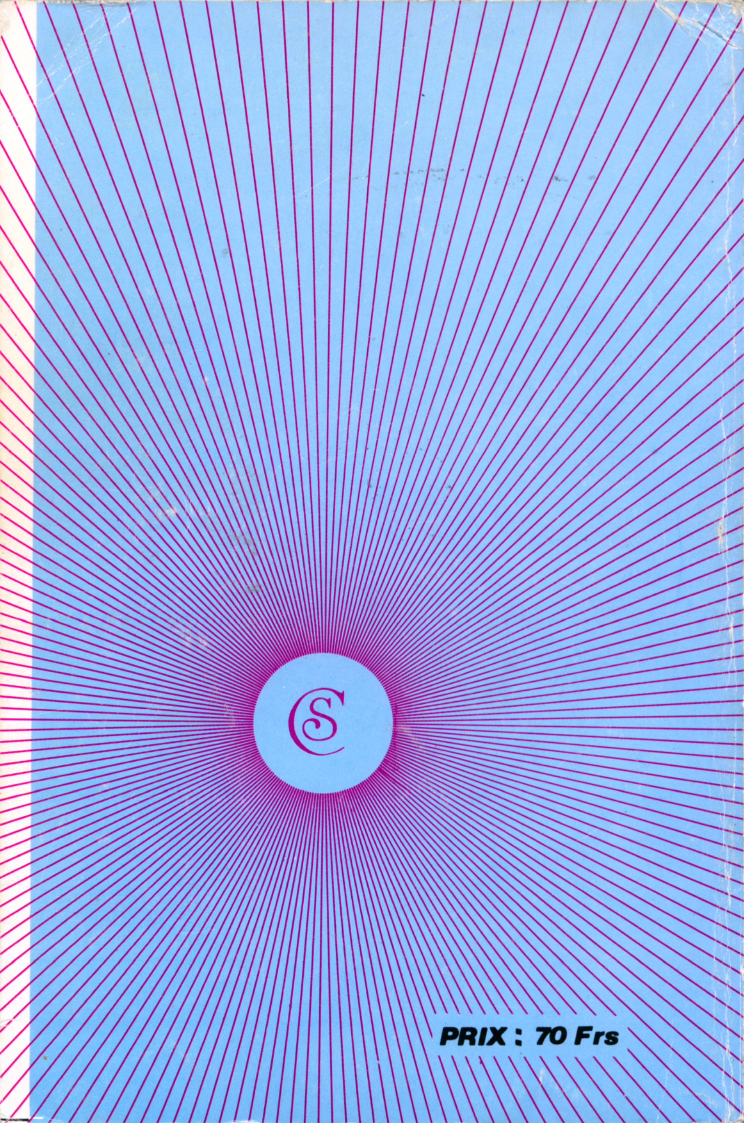
```
40:FOR i=1 TO LEN(m$):a=INSTR(a$,MID$(m$
,i,1))+1:e=ASC(MID$(c$,a))-32-64*(a>37):
GOSUB 40:NEXT:e=42:GOSUB 40:PRINT:PRINT:
PRINT"Encore (O-N) ? ";:WHILE i$="":i$=U
PPER$(INKEY$):WEND:PRINT i$:IF i$="O" TH
EN RUN ELSE END
40 e$=MID$(BIN$(e),2):FOR j=1 TO LEN(e$)
:b$=MID$(e$,j,1):IF b$="1" THEN PRINT"_"
;:SOUND 1,s,3*d,15 ELSE IF b$="0" THEN P
RINT".";:SOUND 1,s,d,15 ELSE PRINT" ":SO
UND 1,s,4*d,0
50 SOUND 1,s,d,0:WHILE SQ(1)>127:WEND:NE
XT j:PRINT" ";:SOUND 1,s,2*d,0:WHILE SQ(
1)>127:WEND:RETURN
```

```
1 WINDOW #1,1,40,1,1:RANDOMIZE TIME:n=IN
T(RND(1)*8)+2:FOR i=1 TO 8 STEP 2:j=INT(
i/2)+6:a$=CHR$(n+48)+"x"+CHR$(j+48):GOSU
B 11:a$=STR$(n*j):GOSUB 11:NEXT:CLS:k=2:
GOSUB 7
3 INPUT#1,1,m:k=1:GOSUB 7:IF e(1)=e(m) T
HEN q=q+1:e(1)=0:e(m)=0:GOTO 5
4 k=2:GOSUB 7
5 IF q=4 THEN END
6 GOTO 3
7 c=0:FOR y=300 TO 200 STEP -100:FOR x=9
6 TO 336 STEP 80:MOVE x,y:c=c+1:IF e(c)=
0 THEN 10
8 IF k=2 THEN FOR i=1 TO 34:DRAWR 0,-80,
1:MOVER 1,0:DRAWR 0,80:MOVER 1,0:NEXT:LO
CATE x/16+3,(400-y)/16+3:IF e(c)<>0 THEN
 PRINT USING "#";c
9 IF k=1 THEN IF c=1 OR c=m THEN MOVER 2
,-2:FOR i=1 TO 32:DRAWR 0,-76,0:MOVER 1,
0:DRAWR 0,76:MOVER 1,0:NEXT:LOCATE x/16+
2,(400-y)/16+3:PRINT t$(c)
10 NEXT:NEXT:RETURN
11 h=INT(RND(2)*8)+1:IF t$(h)<>""THEN 11
12 t$(h)=a$:e(h)=n*j:RETURN
```

Composition : SORACOM
Maquette : SORACOM
Impression : VANDEN BRUGGE
N° d'éditeur : 0 58

ISBN 2-904032-47-9

Illustration de couverture tirée de THANATOS logiciel édité par UBI-Soft.


PRIX : 70 Frs

COMPILATION DES NUMEROS 1··2··3··4